

# LP1 – Esame

*prof. Piero Bonatti*

*prof. Marcello Sette*

5 luglio 2010

## Domande generali – max 6 punti

### Esame 5/7/2010 – Domande generali

#### Domanda 1

Dato il codice C:

```
int x, *y;  
y = &x;
```

Barrare l'interpretazione corretta della espressione `&x` nell'assegnamento:

- [ ] `env(x)`      [ ] `mem(env(x))`      [ ] `env(mem(x))`      [ ] `mem(mem(env(x)))`

(2 punti)

#### Domanda 2

Barrare tutte le affermazioni vere:

- [ ] SQL è un linguaggio di programmazione general purpose.  
[ ] Nel paradigma funzionale puro non c'è memoria, solo ambiente.  
[ ] Nel paradigma funzionale puro non esistono cicli, ma solo ricorsione.  
[ ] Nel paradigma logico non c'è distinzione netta tra input e output.  
[ ] Il primo FORTRAN non ha gestione dinamica della memoria.

(2 punti)

#### Domanda 3

Disegnare il data object relativo alla dichiarazione in C: `int *x = &y;`

(2 punti)

## Esercizio 1: Paradigma imperativo – max 11 punti

Qual è l'output di questo programma nelle seguenti modalità:

- scoping dinamico, [MODE] = IN x copia
- scoping statico, [MODE] = IN x copia
- scoping statico, [MODE] = IN x riferimento
- scoping statico, [MODE] = IN OUT x copia

```
program p1
int a; int b; int c; int d;
procedure p2([IN OUT x rif] int d,[IN OUT x copia] int a)
int c;
BEGIN
c=d;
d=2;
if b=6 then a=b else a=c*3;
b=4;
p3(c, d);
write(a, b, c, d);
END

procedure p3([IN OUT x rif] int a,[MODE] int d)
int b;
procedure p4([IN OUT x rif] int b,[IN OUT x rif] int d)
int c;
BEGIN
c=b+4;
b=1;
d=3;
a=3;
write(a, b, c, d);
END

BEGIN
b=d;
a=b*4;
d=a;
c=2;
p4(d, c);
write(a, b, c, d);
END

BEGIN
a=2;
b=0;
c=1;
d=0;
p2(c, b);
write(a, b, c, d);
END
```

## Esercizio 2: UML – max 15 punti

1. Ogni azienda produce un solo bene e ha altre aziende (max 7) tra i propri fornitori.

*Produrre un diagramma di classe per rappresentare il testo precedente (max 2 punti).*

2. Un allevamento è una azienda. PollaioFelice è un allevamento che produce uova.

*Identificare oggetti e classi; riscrivere, eventualmente integrandolo o modificandolo, il diagramma precedente (max 1 punto).*

3. Un consumatore acquista un bene da una azienda e lo usa. Ogni bene ha un suo modo particolare di essere usato (per esempio, usare le uova significa mangiarle, mentre usare una penna significa scrivere). Inoltre, ogni particolare azienda ha un modo proprio di produrre il bene (cioè un allevamento di galline produce il proprio bene in modo diverso da un mobilificio).

*Riscrivere, eventualmente integrandolo o modificandolo, il diagramma precedente, specificando le classi e gli eventuali metodi astratti (max 3 punti).*

4. Un bene può essere usato parzialmente.

*Riscrivere, il diagramma precedente per caratterizzare meglio i beni, specificandone la quantità residua (un intero) e modificando opportunamente i metodi. Commentare il diagramma, specificando, se possibile, le visibilità di attributi e metodi e gli eventuali metodi astratti (max 4 punti).*

5. Marcello è un consumatore che acquista una scatola di dieci uova dal PollaioFelice. Egli ne usa (mangia) uno.

*Produrre un diagramma di sequenza per rappresentare lo scenario precedente in modo che sia compatibile con l'ultimo dei diagrammi di classe prodotti (max 2 punti).*

6. Un consumatore deve avere solo la possibilità di usare, ma non quella di produrre un bene.

*Utilizzando in modo opportuno le interfacce, modificare l'ultimo diagramma di classe precedentemente prodotto, in modo da rispettare il nuovo vincolo (max 3 punti).*

## Esercizio 1: Paradigma imperativo – Soluzioni

```
***** Mode: in_copia
***** Scoping: dynamic
3 1 12 3
3 2 3 1
3 4 3 2
2 3 2 0
```

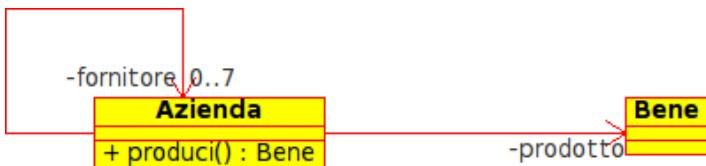
```
***** Mode: in_copia
***** Scoping: static
3 1 12 3
3 2 3 1
3 4 3 3
2 3 3 0
```

```
***** Mode: in_rif
***** Scoping: static
Compile time error:
Illegal assignment d=a to IN/ref parameter of p3
```

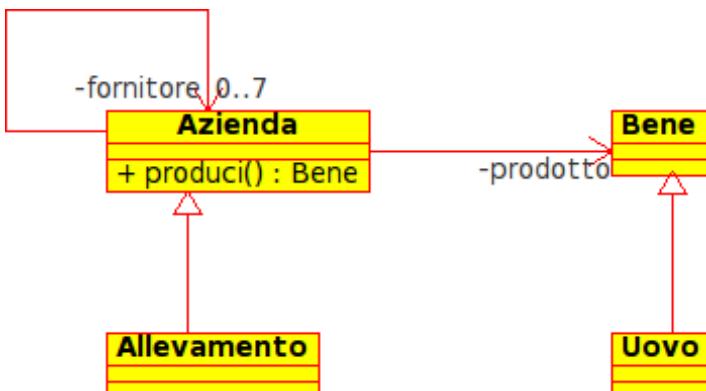
```
***** Mode: in_out_copia
***** Scoping: static
3 1 12 3
3 2 3 1
3 4 3 1
2 3 1 0
```

## Esercizio 2: UML – Soluzioni

1

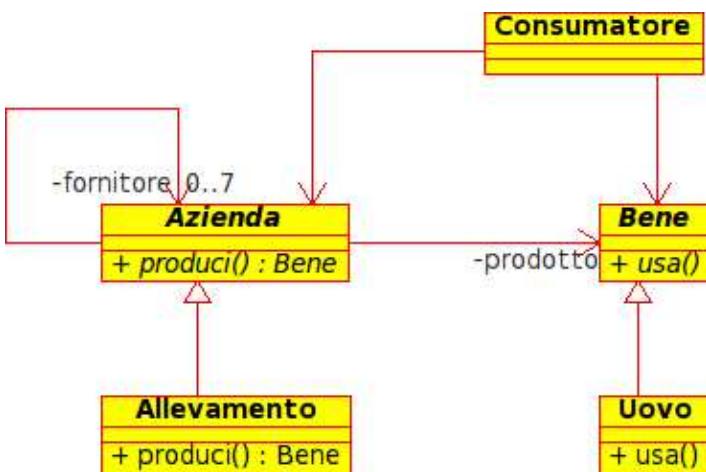


2

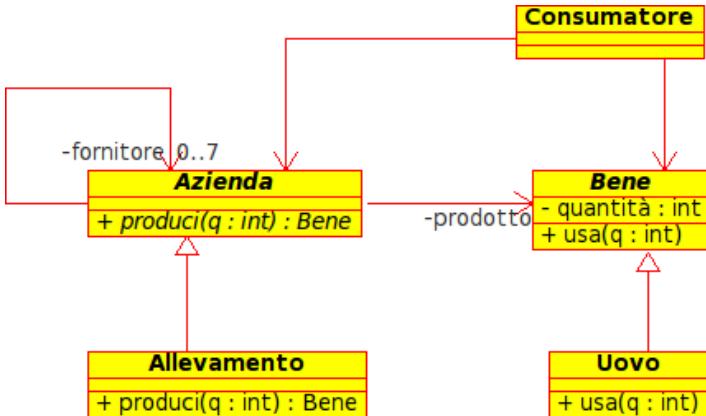


PollaioFelice è una istanza di Allevamento.

3



4

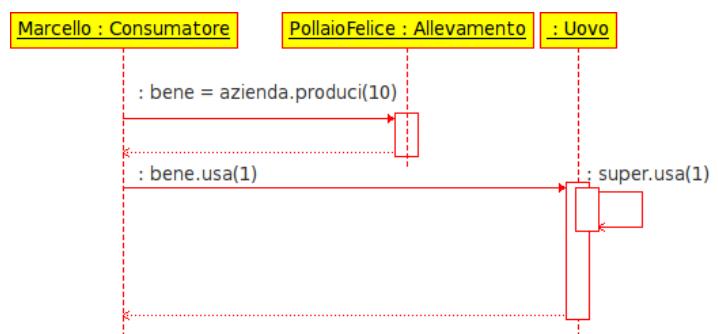


In questo caso, se si sceglie di aggiungere un attributo privato alla classe **Bene**, allora il metodo **usa**, nella stessa classe, non può più essere astratto, ma deve contenere il codice adatto a manipolare quell'attributo (il codice non può risiedere nella sottoclass, visto che l'attributo è privato). Il metodo sovrapposto nella sottoclasse, quello che consente l'uso in modo particolare del bene, è però quello che sarà direttamente invocato (polimorfismo); esso dovrà adoperare, per decurtare la quantità, il metodo della superclasse.

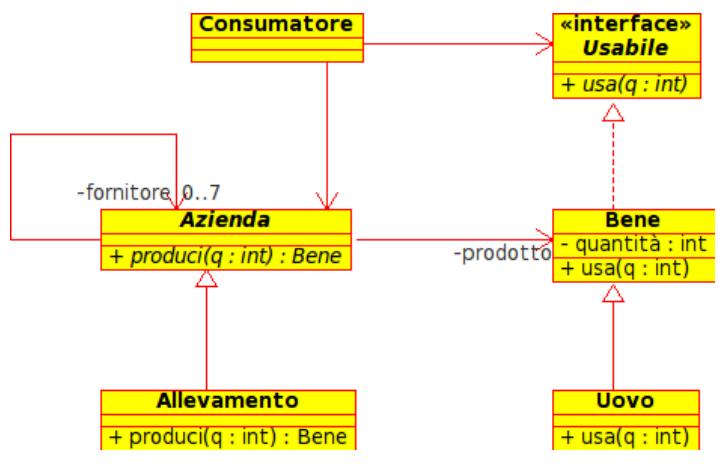
Diverso è se si scegliesse di aggiungere l'attributo privato alla classe **Uovo**, perché ogni altra futura sottoclasse di **Bene** dovrà avere analogo attributo ed analogo brano di codice in **usa()**.

Ancora peggio se si scegliesse di dare visibilità superiore all'attributo nella classe **Bene**, perché in questo caso si romperebbe l'incapsulazione di quella classe.

5



6



# Esame di LP1

Prof Piero Bonatti

20 Settembre 2011

## Domande generali – Max 6 punti

**Esercizio 1:** Barrare tutte le risposte corrette.

Dire se il polimorfismo parametrico puro (quello dei *template*):

- a) E' più adatto del polimorfismo per inclusione per rappresentare insiemi di oggetti omogenei  
SI [ ] NO [ ]
- b) E' più adatto del polimorfismo per inclusione per rappresentare insiemi di oggetti eterogenei  
SI [ ] NO [ ]
- c) Ammette il controllo di tipi completo a tempo di compilazione  
SI [ ] NO [ ]
- d) Può generare errori di tipo a runtime  
SI [ ] NO [ ]

[2 punti]

**Esercizio 2:** Nell'assegnamento  $x = *y$  dei linguaggi C e C++, la parte destra (cioè l'espressione  $*y$ ) denota:

- a)  $mem(y)$  [ ]
- b)  $mem(env(y))$  [ ]
- c)  $env(mem(y))$  [ ]
- d)  $mem(mem(env(y)))$  [ ]  
~~X~~

[2 punti]

**Esercizio 3:**

Disegnare i data object relativi al seguente codice C:

```
int x = 1;  
int * y = &x;  
int * z = y;
```

[2 punti]

## Esercizio sul passaggio di parametri – Max 11 punti

Dire qual è l'output del seguente programma nei casi elencati qui sotto:

1. Scoping dinamico, [MODE] = IN per copia
2. Scoping dinamico, [MODE] = OUT per copia
3. Scoping dinamico, [MODE] = OUT per riferimento
4. Scoping statico, [MODE] = IN per riferimento
5. Scoping statico, [MODE] = OUT per copia

Mostrare gli stack di attivazione (pena la perdita di punti), tranne nei casi di errore.

```
program p1
int a; int b; int c; int d;
procedure p2([IN OUT x rif] int d)
int c;
BEGIN
c=a;
if d=50 then d=c+4 else d=a*4;
b=c+3;
p4(b);
write(a, b, c, d);
END

procedure p3([IN OUT x rif] int a,[MODE] int b)
int d;
BEGIN
d=a*1;
a=4;
# ← b=1;
c=3;
write(a, b, c, d);
END

procedure p4([IN OUT x rif] int d)
int c;
BEGIN
c=b;
d=4;
b=a+1;
a=2;
p3(a, c);
write(a, b, c, d);
END

BEGIN
a=4;
b=0;
c=4;
d=3;
p2(d);
write(a, b, c, d);
END
```

1. **Esercizio 1.** Disegnare un diagramma delle classi per la seguente applicazione:

Una pay TV digitale mantiene

1. Un insieme di clienti, di cui conserva nome, indirizzo e codice fiscale;
2. Un insieme di canali, caratterizzati dal nome e da un responsabile, di cui si specificano nome, indirizzo e codice fiscale;
3. Un insieme di programmi, con il giorno e l'ora di trasmissione, il loro titolo e una descrizione testuale.

Ogni canale è associato ai suoi programmi del mese corrente. Ogni cliente ha un abbonamento, con un inizio e una scadenza, che specifica quali canali possono essere visti dal cliente.

Il sito della pay TV può creare una connessione tra un cliente e un canale per permetterne la visione. Ogni cliente può avere al massimo una connessione attiva.

[max 9 punti]

2. **Esercizio 2.** Disegnare un diagramma di sequenza per il seguente scenario, badando ad inserire nel diagramma delle classi i metodi necessari.

1. Un cliente chiede alla pay TV di vedere il canale 77;
2. La pay TV verifica che il canale sia tra quelli accessibili dal cliente;
3. La pay TV vede che il cliente ha già una connessione attiva e la elimina;
4. La pay TV crea una connessione per il canale 77 e la rende disponibile al cliente.

[max 6 punti]

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

## Prova n. 1

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection al termine dell'esecuzione del metodo *main*?

```
public class LEDTelevision {
    static String marca= new String();
    int numAttacchi;
    static void setParametri(int numcanali) {
        int numCanali=numcanali;
        LEDTelevision t = new LEDTelevision();
    }
    public static void main(String [] argv) {
        LEDTelevision.setParametri(213);
        LEDTelevision.setParametri(514);
    }
}
```

- A. 2
- B. 5
- C. 7
- D. 3
- E. 6

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv) {
        try {
            n();
            System.out.print(1);
        }
        catch( MyExc2 s ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc1 j ) {
            throw( new Error() );
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

```
static void n() {
    try {
        System.out.print(4);
    }
    catch( MyExc3 u ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
    }
}
```

- A. 41
- B. Errore a tempo di compilazione
- C. 4613222222... (ciclo infinito)
- D. 46132Exception in thread main MyExc3
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc3 x ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( MyExc2 i ) {
            System.out.print(4);
        }
        catch( MyExc1 u ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

```

}
static void q() throws Exception {
    try {
        if(false) throw new MyExc1();
        if(false) throw new MyExc3();
        System.out.print(7);
    }
    catch( MyExc3 u ) {
    }
    catch( MyExc1 s ) {
        System.out.print(8);
    }
}

```

- A. Errore a tempo di compilazione  
B. 17263Exception in thread main java.lang.Exception  
C. 1726Exception in thread main MyExc3  
D. 172  
E. Nessuna delle precedenti
- 

4. Dire quale delle seguenti affermazioni è falsa:

- A. Quando due metodi hanno lo stesso nome non sempre si ha overloading  
B. Nell'overloading due metodi non possono avere lo stesso nome ma diverso tipo di parametri  
C. La lunghezza di un array non può essere variata dopo la sua creazione  
D. Un array ha una sola superclasse  
E. Una classe può essere dichiarata final
- 

5. Quale dei seguenti casi è un overloading scorretto del metodo int m(String s, long id) {}

- A. int m(char c, long id) {}  
B. int m(String s) {}  
C. char m(String s, int id) {}  
D. char m(String s, long id) {}  
E. char m(char c) throws Exception {}
- 

6. Date le dichiarazioni:

```

Object g;
Error m;
String v;
g = new Error();
v = new String("");
m = new Error();

```

indicare quale dei seguenti assegnamenti è corretto a tempo di esecuzione.

- A. v = (String) g;  
B. v = (String) m;  
C. m = (Error) v;  
D. m = (Error) g;  
E. Nessuno dei precedenti
- 

7. Qual è l'output di questo codice??

```

class Triangolo {
    private double base;
    public double altezza;
    void setdim(double b, double h) {
        base = b;
        altezza = h;
    }
}
class TriangoloEquilatero extends Triangolo{
    protected double area;
    void area() {
        area = (base*altezza)/2;
    }
    public static void main (String args[]) {
        TriangoloEquilatero te=
        new TriangoloEquilatero();
        te.setdim(4,2);
        te.area();
        System.out.println(te.area());
    }
}

```

- A. 4.0  
B. 8.0  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```

public class Conto {
    String codice;
    public void setCodice(String codice) {
        this.codice=codice;
    }
    public String getCodice(){
        return codice;
    }
    public static void main(String [] argv) {
        setCodice("abc");
        System.out.print(codice);
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione

- C. null
- D. codice
- E. abc

---

9. Quale è l'output di questo codice?

```
interface Albero{  
    int setnumRami(int numR){  
        System.out.println("Albero");  
        return num;  
    }  
}  
  
class Quercia implements Albero{  
    int numRami=12;  
    public int setnumRami(int num){  
        System.out.println("Quercia");  
        return num;  
    }  
    public static void main(String args[]){  
        Quercia q= new Quercia();  
        q.numRami=q.setnumRami(5);  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Albero  
5
- D. Quercia  
12
- E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

```
public class Balena{  
    String nome;  
    boolean viveLibero;  
    Balena() {  
        this.nome="Nemo";  
        this.viveLibero=false;  
    }  
    Balena(String nome,  
           boolean viveLibero) {  
        this.nome=nome;  
        this.viveLibero=viveLibero;  
    }  
    protected void printData(){  
        System.out.print(nome+" ");  
        if(this.viveLibero)  
            System.out.println("vive  
                           libero in mare");  
        else  
            System.out.println("vive  
                           in un acquario");  
    }  
    public static void main(String[] argv){
```

```
        Balena b1= new Balena("FreeWilly", true);  
        Balena b2= new Balena(false);  
        b1.printData();  
        b2.printData();  
    }
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. FreeWilly vive libero in mare  
Nemo vive in un acquario
- D. Nemo vive libero in mare  
FreeWilly vive in un acquario
- E. Nessuna delle precedenti

---

11. Qual è l'output di questo codice?

```
class Codice {  
    String codice = "12345";  
  
    Codice(String c){ codice = c; }  
  
    void verificaCodice(String c) {  
        if (c == codice)  
            System.out.print("Codice OK! ");  
        else  
            System.out.print("Codice Errato. ");  
    }  
  
    public static void main(String[] args) {  
  
        String codice = new String("12345");  
        Codice c = new Codice(codice);  
  
        c.verificaCodice("12345");  
        codice.concat("678");  
        c.verificaCodice(codice);  
    }  
}
```

- A. Errore a compile time
- B. Codice OK! Codice Errato.
- C. Codice Errato. Codice OK!
- D. Codice OK! Codice OK!
- E. Codice Errato. Codice Errato.

---

12. Quale output si ottiene invocando il metodo q?

```
class L {  
    private String s4 = "abcd";  
    private String s5 = s4.concat("e");  
    private String s7 = new String("abc");  
    void q() {  
        String s3 = new String("ab");  
        p(s3, s3, s5, s7);  
    }  
    void p(String s1, Object s2,
```

```

        String s6, Object s8) {
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s6 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s8 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
  - B. 001
  - C. 111
  - D. 010
  - E. 000
- 

13. Qual è l'output di questo codice?

```

class Poligono{
    int num_lati;
    Poligono(int numlati) {
        num_lati=numlati;
        System.out.println("Poligono:
                           Numero Lati " + num_lati);
    }
}
class Quadrato extends Poligono{
    int lunghezza;
    Quadrato()
    {
        super(4);
        this.lunghezza=25;
        System.out.println("Quadrato:
                           Numero Lati " + num_lati);
    }
}
public static void main (String[] agrv){
    Quadrato q = (Quadrato) new Poligono(5);
}

```

- A. Errore a tempo di esecuzione
  - B. Errore a tempo di compilazione
  - C. Quadrato: Numero Lati 4
  - D. Poligono: Numero Lati 4
   
Quadrato: Numero Lati 4
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
            System.out.print(2);
        }
        catch( MyExc2 t ) {
            System.out.print(3);
        }
        catch( Exception j ) {
            throw( new MyExc3() );
        }
        catch( MyExc1 c ) {
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc3 a ) {
        }
        catch( Exception c ) {
        }
        finally {
            throw( new MyExc1() );
        }
    }
}

```

- A. 14
  - B. 14Exception in thread main MyExc3
  - C. Errore a tempo di compilazione
  - D. 142
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

class Computer {
    String tipoCPU;
    protected String dimRAM;
    void setTipoCPU(String tipoCPU) {
        this.tipoCPU=tipoCPU;
    }
}
class NetBook extends Computer{
    int polliciMonitor;
    String marca="Intel";
    void print(){
        setTipoCPU("Atom");
        System.out.println(marca+" "+tipoCPU);
    }
}

```

```
}

public static void main (String args[]) {
    NetBook n= new NetBook();
    n.print();
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Intel Atom
- D. Atom Intel
- E. Nessuna delle Precedenti

```
String d;
Integer e;
Object z;
```

indicare quale dei seguenti assegnamenti è corretto a tempo di compilazione.

- A. d = z;
- B. d = (String) e;
- C. d = e;
- D. e = z;
- E. z = (Object) e;

16. Date le dichiarazioni:

---

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 3

1. Qual è l'output di questo codice?

```
class Trespolo extends Tavolo{
    Trespolo(int nump, String mater) {
        super.numpiedi=nump;
        super.materiale=mater;
    }
}
class Tavolo{
    private int numpiedi;
    protected String materiale="";
    Tavolo() {};
    Tavolo(int nump, String materiale) {
        this.numpiedi=nump;
        this.materiale=materiale;
    }
    void print(){
        System.out.print("Tavolo con
        "+numpiedi+" gambe ");
        System.out.println(" e di "+materiale);
    }
    public static void main(String args[]) {
        Trespolo t=new Trespolo(3,"Legno");
        t.print();
    }
}
```

- A. Tavolo con 0 gambe e di Legno
- B. Errore a tempo di compilazione
- C. Tavolo con 3 gambe e di Legno
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

- B. s = (Integer) g;
- C. r = (Error) s;
- D. s = (Integer) r;
- E. Nessuno dei precedenti

3. Qual è l'output di questo codice?

```
interface FormaGeometrica{
    void calcola_perimetro();
    void calcola_area();
}
class Triangolo implements FormaGeometrica{
    int latol=1;
    int lato2=2;
    int lato3=3;

    public void calcola_perimetro(){
        int p=latol+lato2+lato3;
        System.out.println("Perimetro= "+p);
    }
    public void calcola_area(int altezza){
        //int altezza=5;
        int a=(lato3*altezza)/2;
        System.out.println("Area= "+a);
    }
}
public static void main(String args[]){
    Triangolo t=new Triangolo();
    t.calcola_perimetro();
    t.calcola_area(5);
}
```

- A. Perimetro= 6  
Area= 7
- B. Perimetro= 6  
Area= 4
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Object g;
Error r;
Integer s;
s = new Integer(0);
r = new Error();
g = new Integer(0);
```

indicare quale dei seguenti assegnamenti è corretto a tempo di esecuzione.

- A. r = (Error) g;

4. Qual è l'output di questo codice?

```

class TavolinoDaSalotto extends Tavolo2{
    TavolinoDaSalotto(int nump, String mater) {
        super(nump, mater);
    }
}
class Tavolo2{
    private int numpiedi;
    protected String materiale="";
    Tavolo2(int nump, String materiale){
        this.numpiedi=nump;
        this.materiale=materiale;
    }
    void print(){
        System.out.print("Tavolo con
            "+numpiedi+" gambe ");
        System.out.println("e di "+materiale);
    }
}
public static void main(String args[]){
    TavolinoDaSalotto t=new
        TavolinoDaSalotto(4, "Vetro");
    t.print();
}
}

```

- A. Tavolo con 0 gambe e di Vetro
  - B. Errore a tempo di compilazione
  - C. Tavolo con 4 gambe e di Vetro
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```

Integer a;
Exception n;
Object w;

```

indicare quale dei seguenti assegnamenti è corretto  
a tempo di compilazione.

- A. w = a;
  - B. a = w;
  - C. n = (Exception) a;
  - D. a = n;
  - E. n = w;
- 

6. Qual è l'output di questo codice?

```

public class Persona {
    String nome;
    Integer eta;
    public Persona(String nome) {
        this.nome=nome;
    }
}

```

```

}
public Persona(String nome, Integer eta) {
    this.nome;
    this.eta = eta;
}
protected void printData(){
    System.out.print("Nome: " + nome);
    System.out.print(" - Eta': " + eta);
}
public static void main(String [] argv) {
    Persona p1 = new Persona();
    Persona p2 = new Persona("Antonio");
    p1.printData();
    p2.printData();
}
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. Nome: Antonio - Eta': 0
  - D. Nome: Antonio - Eta': null
  - E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Tutti i tipi di eccezioni estendono la classe Object
  - B. Un array con riferimento null ha lunghezza zero
  - C. Quando due metodi hanno lo stesso nome si ha overloading
  - D. Un array non possiede dei membri
  - E. Nell'overloading due metodi non possono avere lo stesso nome ma diverso tipo di parametri
- 

8. Quale output si ottiene invocando il metodo q?

```

class B {
    private Float f1 = new Float(50.0);
    private Boolean b1 = new Boolean(true);
    Boolean b2;
    private Integer i2 = new Integer(5);
    void q() {
        b2 = b1;
        m(f1);
    }
    void m(Object f2) {
        Integer i1;
        i1 = new Integer(5);
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 101  
 B. 011  
 C. 010  
 D. 110  
 E. 000
- 

9. Qual è l'output di questo codice?

```

abstract class FormaGeometrica{
    void calcola_perimetro(){}
    void calcola_area(){}
}
class Quadrato extends FormaGeometrica {
    int lato=2;
    public void calcola_perimetro(){
        System.out.println("Perimetro= "+lato*4);
    }
    public void calcola_area(){
        System.out.println("Area= "+(lato*lato));
    }
    public static void main(String args[]){
        Quadrato q=new FormaGeometrica();
        q.calcola_perimetro();
        q.calcola_area();
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. Perimetro= 6 Area= 4  
 D. Perimetro= 6 Area= 0  
 E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

abstract class Fiore{
    static final int numPetali = 6;
}
class Rosacee extends Fiore{
    static int numSpine = 12;
    Rosacee(){
        System.out.println("Famiglia Rosacee");
    }
}

```

```

        }
    }
    class Tiliacee extends Fiore{
        String colore = "bianco";
        Tiliacee(){
            System.out.println("Famiglia Tiliacee");
        }
    }
    class Rosa extends Rosacee {
        static final int numPetali = 8;
        Rosa() {
            System.out.println("Rosa-Fiore
                :" + numPetali + " petali");
        }
    }
    public static void main(String [] argv){
        Fiore r1 = new Rosa();
        Fiore r2 = new Rosacee();
        Tiliacee r3 = new Rosa();
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. Famiglia Rosacee  
 Rosa-Fiore :8 petali  
 Famiglia Rosacee  
 Famiglia Tiliacee  
 D. Famiglia Rosacee  
 Rosa-Fiore :6 petali  
 Famiglia Rosacee  
 Famiglia Tiliacee  
 E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 r ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void q() {
        try {
            throw( new Error() );
        }
        catch( Error t ) {
            }
        }
    }
}

```

```

        finally {
            System.out.print(3);
        }
    }
}

```

- A. 1  
 B. 13222222... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 132Exception in thread main MyExc1  
 E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc1 s ) {
        }
        catch( MyExc2 y ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
            throw( new MyExc2() );
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 u ) {
            System.out.print(3);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
}

```

- A. 342Exception in thread main MyExc2  
 B. 31  
 C. 3421  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B2 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( Exception k ) {
            System.out.print(3);
        }
        catch( MyExc1 u ) {
            System.out.print(4);
        }
        catch( MyExc3 k ) {
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(6);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 15632Exception in thread main MyExc1  
 C. 1562  
 D. 1563Exception in thread main MyExc1  
 E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

interface FormaGeometrica{
    void calcola_perimetro();
    void calcola_area();
}

class Quadrato2 implements FormaGeometrica {
    int lato=2;
    public void calcola_perimetro(){
        System.out.println("Perimetro= "+lato*4);
    }
    public void calcola_area(){
        System.out.println("Area= "+(lato*lato));
    }
    public static void main(String args[]){
        Quadrato2 q=new Quadrato2();
        q.calcola_perimetro();
        q.calcola_area();
    }
}

```

```
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. Perimetro= 6 Area= 4
- D. Perimetro= 6 Area= 0
- E. Nessuna delle precedenti

---

15. Quanti oggetti possono essere deallocati mediante garbage collection nel punto indicato ~~e non prima?~~  
**AL MASSIMO**

```
class Sirio extends Stella{  
    String nome="Sirio";  
}  
  
public class Stella {  
    String nome = "";  
  
    public static void main(String [] argv){  
        Sirio s1=new Sirio();  
        Stella s2=new Stella ();  
        s1=s2;  
        s2=null; /* QUI */  
    }  
}
```

- A. 0
- B. 1
- C. 2
- D. 3
- E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

```
public class Docente {  
    protected String nome="Bianco";  
    protected double Id_Docente=2323 ;  
    protected String materia="Storia";  
  
    public String print() {  
        return " Nome: " + nome + " " +  
            " Materia :" + materia;  
    }  
  
    public static void main(String args[]) {  
  
        Docente d1=new Docente();  
        DocenteUniverisitario d2=new Docente();  
        System.out.println(d1.print());  
        System.out.println(d2.print());  
    }  
}  
  
class DocenteUniverisitario extends Docente {  
    protected String corsodiLaurea="Filosofia" ;  
    public String print() {  
        return " Nome: " + nome + " " +  
            " Materia :" + materia + " " +  
            " Corso Di Laurea :" + corsodiLaurea;  
    }  
}
```

- A. Errore a tempo di esecuzione
- B. Nome: Bianco Materia :Storia  
Nome: Bianco Materia :Storia Corso Di Laurea  
:Filosofia
- C. Nome: Bianco Materia :Storia  
Nome: Bianco Materia :Storia Corso Di Laurea  
: null
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 1**

1. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

2. Qual è il *sovraffaccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
```

```
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
```

```

String s8 = "";
s2 = s1.concat("abcd");
s6 = "";
a4 = a3;
q(s4, s6, s8, a4, a3);
}
void q(String s3, String s5, String s7,
    float[] a1, Object a2){
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s5 == s7) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(a2 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

5. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

6. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {

```

```

        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
A. 1
B. 2
C. 3
D. 4
E. 6

```

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

8. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

---

9. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

---

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
```

```
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

---

11. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

---

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
```

```

    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**15. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";
}

```

```

public void m() {
    System.out.print(super.s
        + ((X) this).s
        + ((Y) this).s
        + this.s);
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

**16. Qual'è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {

```

```

        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 2

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
            + ((X) this).s  
            + ((Y) this).s  
            + this.s);  
    }  
  
    public static void main(String[] args) {  
        new Y().m();  
    }  
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Dato un file contenente il seguente codice:

```
package pk;  
  
public class A {  
    public class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

3. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
  
    class Skyscraper extends Building {  
        Skyscraper(int f, String c){super(f,c);}  
  
        public static void print() {  
            System.out.print(floors + " "  
                + constructor + " ");  
        }  
    }  
}
```

```

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

A. Errore a tempo di compilazione

- B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

#### 6. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

#### 7. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

9. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {

```

```

Building a = new Building(3,"Jones");
Building b = new Skyscraper(21,"Smith");
Skyscraper c = new Skyscraper(50,"Black");
a.print();
b.print();
c.print();
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 63Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception

- C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

A. XXYY
B. XYYY
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {

```

```

        System.out.print(floors + " "
                        + constructor + " ");
    }

}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**14. Date le dichiarazioni:**

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;

- C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

**15. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?**

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

**16. Dire quale delle seguenti affermazioni è falsa:**

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 3**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}
```

```

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
 B. Jones 21 Smith 50 Black  
 C. Black Black 50 Black  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");

```

```

        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
 B. Jones 21 Smith 50 Black  
 C. Black Black 50 Black  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

5. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.  
 B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
 C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().  
 D. Un'istanza della classe I può essere costruita con new A().new I().  
 E. Nessuna delle precedenti.
- 

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {

```

```

try {
    System.out.print(3);
    throw( new MyExc3() );
}
finally {
    System.out.print(4);
    throw( new MyExc3() );
}
}
}

```

- A. 34  
B. 342Exception in thread "main" MyExc3  
C. Errore a tempo di compilazione  
D. 342  
E. Nessuna delle precedenti
- 

7. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)  
B. public byte f(Object y)  
C. protected int f(String z)  
D. int f(String s) throws Exception  
E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XXYY  
B. XYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

9. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

- ```

public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A() {}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.

```
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            ()
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
    }
}

```

```

        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 152  
 C. 1524  
 D. 154  
 E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
 B. Jones 21 Smith 50 Black  
 C. Black Black 50 Black  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

12. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;  
 B. h = (C1) a;  
 C. h = (C1) t;  
 D. a = (Sub2) h;  
 E. Nessuno dei precedenti
- 

13. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

---

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

---

15. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;

E. w = (Object [] []) z;

---

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    int ml = 20;  
    Coffee() {  
        System.out.println(brand);  
    }  
    Coffee(int ml){  
        this.ml = ml;  
        System.out.println(ml);  
    }  
    public static void main(String args[]){  
        Coffee ristretto = new Coffee();  
        Coffee kazar = new Coffee();  
        Coffee vanillio = new Coffee(40);  
        ristretto = null;  
        vanillio = ristretto;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 4**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args){
    new Y().m();
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

4. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    String constructor;  
    int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
}
```

```
}

public void print() {
    System.out.print(constructor + " ");
}
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
 B. Jones 21 Smith 50 Black  
 C. Black Black 50 Black  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. 6
- 

9. Qual è il *sovraffaticamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}  
 B. void A(){}  
 C. private A(int i, String s) {}  
 D. public Object A(int i, String s) {}  
 E. Il costruttore di una classe non può essere sovraccaricato.
- 

10. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;  
 B. h = (C1) a;  
 C. h = (C1) t;  
 D. a = (Sub2) h;  
 E. Nessuno dei precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
}

```

```

try {
    throw( new MyExc2() );
}
catch( MyExc1 t ) {
}
catch( Exception i ) {
}
catch( MyExc3 s ) {
    throw( new MyExc3() );
}
finally {
    System.out.print(5);
}
}
}

```

- A. Errore a tempo di compilazione  
 B. 152  
 C. 1524  
 D. 154  
 E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
}

```

- A. XXYY  
 B. XYYY  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

13. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

16. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
```

```
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float [] a1, Object a2) {
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

Prova n. 5

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
abstract class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
            + ((X) this).s  
            + ((Y) this).s  
            + this.s);  
    }  
  
    public static void main(String[] args) {  
        new Y().m();  
    }  
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    String constructor;  
    int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public void print() {  
        System.out.print(constructor + " ");  
    }  
  
    class Skyscraper extends Building {  
        Skyscraper(int f, String c) {super(f,c);}  
  
        public void print() {  
            System.out.print(floors + " "  
                + constructor + " ");  
        }  
    }  
  
    class Main {  
        public static void main(String[] args) {  
            Building a = new Building(3,"Jones");  
            Building b = new Skyscraper(21,"Smith");  
            Skyscraper c = new Skyscraper(50,"Black");  
            a.print();  
            b.print();  
            c.print();  
        }  
    }  
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

5. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

6. Qual è il *sovraffaccamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A() {}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.
```

---

7. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
A. 100
B. 011
C. 111
D. 000
E. 101
```

---

8. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().

- D. Un'istanza della classe I può essere costruita con `new A().new I()`.  
E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
```

```
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args){
    new Y().m();
}
```

A. XXYY  
B. XYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
```

```

    c.print();
}
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

13. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
    }
}

```

```

        }
    finally {
        System.out.print(5);
    }
}

static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception  
B. 63Exception in thread "main" java.lang.Exception  
C. 615  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
        }
    }
}

```

```

        System.out.print(2);
    }
    catch( MyExc2 r ) {
    }
    catch( MyExc1 j ) {
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. Errore a tempo di compilazione  
B. 152  
C. 1524  
D. 154  
E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 6**

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}
```

```
class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

3. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

6. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
    }
    finally {
        System.out.print(2);
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
}

```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

8. Qual è il *sovraffaccamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

9. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

10. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011

C. 111

D. 000

E. 101

---

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

A. Errore a tempo di compilazione

B. 152

C. 1524

D. 154

E. Nessuna delle precedenti

---

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)

```

```

        throws Exception {
try {
    p();
    System.out.print(1);
}
catch( MyExc1 e ) {
    System.out.print(2);
}
catch( MyExc3 d ) {
    System.out.print(3);
    throw( new Exception() );
}
catch( Exception i ) {
    System.out.print(4);
}
finally {
    System.out.print(5);
}
}

static void p() throws Exception {
try {
    System.out.print(6);
    if (true) throw( new MyExc3());
    throw( new MyExc1() );
}
catch( MyExc1 t ) {
    throw( new MyExc1() );
}
catch( MyExc2 s ) {
}
}
}

```

- A. 635Exception in thread "main" java.lang.Exception  
B. 63Exception in thread "main" java.lang.Exception  
C. 615  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper(){}
}

```

```

Skyscraper(int f, String c) {super(f,c);}

public static void print() {
    System.out.print(floors + " "
                    + constructor + " ");
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

14. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;  
B. h = (C1) a;  
C. h = (C1) t;  
D. a = (Sub2) h;  
E. Nessuno dei precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 7**

1. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
```

```
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
A. 635Exception in thread "main" java.lang.Exception
B. 63Exception in thread "main" java.lang.Exception
C. 615
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```

---

3. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

4. Qual'è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
-

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(){}  
    Skyscraper(int f, String c) {super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
                        + constructor + " ");  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends Error {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String[] argv){  
        try {  
            q();  
        }  
        catch( MyExc3 x ) {  
        }  
        catch( MyExc1 c ) {  
        }  
    }  
}
```

```
}  
catch( MyExc2 c ) {  
    System.out.print(1);  
}  
finally {  
    System.out.print(2);  
}  
}  
static void q() {  
    try {  
        System.out.print(3);  
        throw( new MyExc3() );  
    }  
    finally {  
        System.out.print(4);  
        throw( new MyExc3() );  
    }  
}  
}  
}
```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.  
                        + ((X) this).s  
                        + ((Y) this).s  
                        + this.s);  
    }  
}  
  
public static void main(String[] args){  
    new Y().m();  
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- 
8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    int ml = 20;  
    Coffee() {  
        System.out.println(brand);  
    }  
    Coffee(int ml) {  
        this.ml = ml;  
        System.out.println(ml);  
    }  
    public static void main(String args[]) {  
        Coffee ristretto = new Coffee();  
        Coffee kazar = new Coffee();  
        Coffee vanillio = new Coffee(40);  
        ristretto = null;  
        vanillio = ristretto;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6
- 

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class A1 {  
    public static void main(String[] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            n();  
            System.out.print(2);  
        }  
        catch( MyExc2 r ) {}  
        catch( MyExc1 j ) {}  
        catch( Exception e ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            throw( new MyExc2() );  
        }  
        catch( MyExc1 t ) {}  
        catch( Exception i ) {}  
        catch( MyExc3 s ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione  
B. 152  
C. 1524  
D. 154  
E. Nessuna delle precedenti
- 

10. Quale output si ottiene invocando il metodo p?

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");  
        String s6;  
        String s8 = "";  
        s2 = s1.concat("abcd");  
        s6 = "";  
        a4 = a3;  
        q(s4, s6, s8, a4, a3);  
    }  
    void q(String s3, String s5, String s7,  
        float[] a1, Object a2){  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s5 == s7) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a2 == a1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 100

- B. 011
- C. 111
- D. 000
- E. 101

---

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c){super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
                        + constructor + " ");  
    }  
}  
  
class Run {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

12. Date le dichiarazioni:

```
class C1 extends A {...}  
class A extends Object {...}  
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;  
C1 h;  
A t;  
t = new C1();  
a = new Sub2();  
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

---

13. Dato un file contenente il seguente codice:

```
package pk;  
  
public class A {  
    public class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

---

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

A. XXYY
B. XYYY
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;
}

```

```

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 8**

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s

```

```
+ ((X) this).s
+ ((Y) this).s
+ this.s);
}

public static void main(String[] args) {
    new Y().m();
}
}

A. XXYY
B. XYYY
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.
```

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

```
    }  
}
```

A. 34

- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

---

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
                        + ((X) this).s  
                        + ((Y) this).s  
                        + this.s);  
    }  
  
    public static void main(String[] args){  
        new Y().m();  
    }  
}
```

- A. XXYY
- B. YYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

5. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;
- C. t = (String []) w;

```
D. w = (Object [] []) t;  
E. w = (Object [] []) z;
```

---

6. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}  
  
A. public A(String b) {}  
B. void A() {}  
C. private A(int i, String s) {}  
D. public Object A(int i, String s) {}  
E. Il costruttore di una classe non può essere sovraccaricato.
```

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper() {}  
    Skyscraper(int f, String c) {super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
                        + constructor + " ");  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black

- C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

9. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
```

```
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}
```

1} E. Nessuna delle precedenti.

---

```
class Run {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

11. Date le dichiarazioni:

```
class C1 extends A {...}  
class A extends Object {...}  
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;  
C1 h;  
A t;  
t = new C1();  
a = new Sub2();  
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

12. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception

13. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

14. Dato un file contenente il seguente codice:

```
package pk;
```

```
public class A {  
    public class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class A1 {  
    public static void main(String[] argv)  
        throws Exception {  
        try {  
            p();  
            System.out.print(1);  
        }  
        catch( MyExc1 e ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 d ) {  
            System.out.print(3);  
            throw( new Exception() );  
        }  
        catch( Exception i ) {  
            System.out.print(4);  
        }  
        finally {
```

```

        System.out.print(5);
    }
}

static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**16. Qual'è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);

```

```

        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 9**

1. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;  
B. h = (C1) a;  
C. h = (C1) t;  
D. a = (Sub2) h;  
E. Nessuno dei precedenti

2. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee(){
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception

- B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends Error { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String[] argv) {  
        try {  
            q();  
        }  
        catch( MyExc3 x ) {  
        }  
        catch( MyExc1 c ) {  
        }  
        catch( MyExc2 c ) {  
            System.out.print(1);  
        }  
        finally {  
            System.out.print(2);  
        }  
    }  
    static void q() {  
        try {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 34
- B. 342Exception in thread "main" MyExc3

C. Errore a tempo di compilazione

- D. 342

- E. Nessuna delle precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    String constructor;  
    int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c) {super(f,c);}  
  
    public void print() {  
        System.out.print(floors + " "  
                         + constructor + " ");  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

7. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}  
  
A. public A(String b) {}  
B. void A() {}  
C. private A(int i, String s) {}
```

- D. public Object A(int i, String s) {}  
E. Il costruttore di una classe non può essere sovraccaricato.
- 

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione  
B. 152  
C. 1524  
D. 154  
E. Nessuna delle precedenti
- 

9. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
```

```
private float [] a3 = new float [8];
private float [] a4;
void p() {
    String s4 = s1.concat("abcd");
    String s6;
    String s8 = "";
    s2 = s1.concat("abcd");
    s6 = "";
    a4 = a3;
    q(s4, s6, s8, a4, a3);
}
void q(String s3, String s5, String s7,
       float[] a1, Object a2){
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 100  
B. 011  
C. 111  
D. 000  
E. 101
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.
                         + ((X) this).s
                         + ((Y) this).s
                         + this.s);
    }
}
```

```

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

14. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

15. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
}
```

```
public static void print() {  
    System.out.print(constructor + " ");  
}  
}  
  
class Skyscraper extends Building {  
    Skyscraper() {}  
    Skyscraper(int f, String c) {super(f, c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
                        + constructor + " ");  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Building a = new Building(3, "Jones");  
        Building b = new Skyscraper(21, "Smith");  
        Skyscraper c = new Skyscraper(50, "Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

Prova n. 10

1. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

2. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

3. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}  
  
A. public A(String b) {}  
B. void A() {}  
C. private A(int i, String s) {}  
D. public Object A(int i, String s) {}  
E. Il costruttore di una classe non può essere sovraccaricato.
```

---

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    String constructor;  
    int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c) {super(f,c);}  
  
    public void print() {  
        System.out.print(floors + " "  
                        + constructor + " ");  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }

```

```

        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

#### 7. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

#### 8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {

```

```

        System.out.print(5);
    }
}

static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception  
B. 63Exception in thread "main" java.lang.Exception  
C. 615  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

9. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee(){
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args){
    new Y().m();
}

```

- A. XXYY  
B. XYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
    }
}

```

```

Skyscraper c = new Skyscraper(50,"Black");
a.print();
b.print();
c.print();
}
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new A.I()`.
  - D. Un'istanza della classe I può essere costruita con `new A().new I()`.
  - E. Nessuna delle precedenti.
- 

13. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {

```

```

            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- }
  - }
  - A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 11**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");    }
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;
```

```

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

#### 4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
    }
}

```

```

        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception  
B. 63Exception in thread "main" java.lang.Exception  
C. 615  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 5. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;  
B. h = (C1) a;  
C. h = (C1) t;  
D. a = (Sub2) h;  
E. Nessuno dei precedenti
- 

#### 6. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)  
B. public byte f(Object y)

- C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

7. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

8. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
```

```
String s = "Y";

public void m() {
    System.out.print(super.s
        + ((X) this).s
        + ((Y) this).s
        + this.s);
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Qual è il sovraccarico corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A() {}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.

```

---

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342

E. Nessuna delle precedenti

---

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 12**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;
```

```
Building(int f, String c) {
    floors = f;
    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}
}

public static void print() {
    System.out.print(floors + " "
                    + constructor + " ");
}
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- 
- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.

- 
- 4. Qual è il *sovraff장imento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}  
  
A. public A(String b) {}  
B. void A() {}  
C. private A(int i, String s) {}  
D. public Object A(int i, String s) {}  
E. Il costruttore di una classe non può essere sovraccaricato.
```

---

- 5. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    int ml = 20;  
    Coffee() {  
        System.out.println(brand);  
    }  
    Coffee(int ml) {  
        this.ml = ml;  
        System.out.println(ml);  
    }  
    public static void main(String args[]) {  
        Coffee ristretto = new Coffee();  
        Coffee kazar = new Coffee();  
        Coffee vanillio = new Coffee(40);  
        ristretto = null;  
        vanillio = ristretto;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

- 
- 6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.  
            + ((X) this).s  
            + ((Y) this).s  
            + this.s);  
    }  
  
    public static void main(String[] args) {  
        new Y().m();  
    }  
}  
  
A. XXYY  
B. XYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
```

---

- 7. Quale output si ottiene invocando il metodo p?

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");  
        String s6;  
        String s8 = "";  
        s2 = s1.concat("abcd");  
        s6 = "";  
        a4 = a3;  
        q(s4, s6, s8, a4, a3);  
    }  
    void q(String s3, String s5, String s7,  
          float[] a1, Object a2) {  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s5 == s7) {  
            System.out.print(1);  
        }  
    }  
}
```

```

    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 100  
 B. 011  
 C. 111  
 D. 000  
 E. 101
- 

#### 8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception

- B. 63Exception in thread "main" java.lang.Exception  
 C. 615  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 152  
 C. 1524  
 D. 154  
 E. Nessuna delle precedenti
- 

#### 10. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione

- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

11. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}
```

```
}
```

```
class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

13. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}
```

```

    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

public void m() {
    System.out.print(super.s
        + ((X) this).s
        + ((Y) this).s
        + this.s);
}

public static void main(String[] args) {
    new Y().m();
}

A. XXYY
B. XYYY
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";
}

```

16. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 13**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}
```

```
}                                }

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

3. Qual è il *sovraffaccimento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

4. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

---

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c){super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
                        + constructor + " ");  
    }  
}
```

```
class Run {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");  
        String s6;  
        String s8 = "";  
        s2 = s1.concat("abcd");  
        s6 = "";  
        a4 = a3;  
        q(s4, s6, s8, a4, a3);  
    }  
    void q(String s3, String s5, String s7,  
          float[] a1, Object a2){  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s5 == s7) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a2 == a1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

---

7. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

---

6. Quale output si ottiene invocando il metodo p?

---

8. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

### 9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

A. Errore a tempo di compilazione

B. 152

C. 1524

D. 154

E. Nessuna delle precedenti

---

### 10. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

### 11. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

### 12. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {

```

```

    }
    catch( MyExc2 c ) {
        System.out.print(1);
    }
    finally {
        System.out.print(2);
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}

```

- A. 34  
 B. 342Exception in thread "main" MyExc3  
 C. Errore a tempo di compilazione  
 D. 342  
 E. Nessuna delle precedenti

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
    }
}

```

```

        b.print();
        c.print();
    }
}

A. Jones Smith 50 Black
B. Jones 21 Smith 50 Black
C. Black Black 50 Black
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
}

A. XXYY
B. XYYY
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {

```

```

        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
        throw( new Exception() );
    }
    catch( Exception i ) {
        System.out.print(4);
    }
    finally {
        System.out.print(5);
    }
}
static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

- 
16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 14**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

2. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2) {
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

3. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno delle precedenti

---

4. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

---

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

6. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
}
```

```

Skyscraper(int f, String c) {super(f,c);}

public static void print() {
    System.out.print(floors + " "
                    + constructor + " ");
}

}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

9. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno

- D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
        }
    }
}

```

```

        System.out.print(1);
    }
    catch( MyExc1 e ) {
        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
        throw( new Exception() );
    }
    catch( Exception i ) {
        System.out.print(4);
    }
    finally {
        System.out.print(5);
    }
}
static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

```

```
public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

15. Qual è il *sovvraccarico* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
```

```
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 15**

1. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
```

- ```
        }
    }
}

A. 635Exception in thread "main" java.lang.Exception
B. 63Exception in thread "main" java.lang.Exception
C. 615
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new A.I()`.
  - D. Un'istanza della classe I può essere costruita con `new A().new I()`.
  - E. Nessuna delle precedenti.
- 

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee(){
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
```

```

        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {

```

```

        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

7. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
    }
}

```

```

}
catch( MyExc2 r ) {
}
catch( MyExc1 j ) {
}
catch( Exception e ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
}
}

static void n() throws Exception {
try {
    throw( new MyExc2() );
}
catch( MyExc1 t ) {
}
catch( Exception i ) {
}
catch( MyExc3 s ) {
    throw( new MyExc3() );
}
finally {
    System.out.print(5);
}
}
}

```

- A. Errore a tempo di compilazione  
B. 152  
C. 1524  
D. 154  
E. Nessuna delle precedenti

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

}
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.

10. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)  
B. public byte f(Object y)  
C. protected int f(String z)  
D. int f(String s) throws Exception  
E. Nessuna delle precedenti.

11. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2) {
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {

```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black

- B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args){
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
- B. h = (C1) a;

- C. `h = (C1) t;`
  - D. `a = (Sub2) h;`
  - E. Nessuno dei precedenti
- 

15. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `t = w;`
  - B. `t = z;`
  - C. `t = (String []) w;`
  - D. `w = (Object [] []) t;`
  - E. `w = (Object [] []) z;`
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
}
```

```
public static void print() {  
    System.out.print(constructor + " ");  
}  
}
```

```
class Skyscraper extends Building {  
    Skyscraper(int f, String c){super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
                        + constructor + " ");  
    }  
}  
  
class Run {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 16

1. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

2. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class A1 {  
    public static void main(String[] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            n();  
            System.out.print(2);  
        }  
        catch( MyExc2 r ) {  
        }  
        catch( MyExc1 j ) {  
        }  
        catch( Exception e ) {  
            System.out.print(3);  
        }  
    }  
}
```

```
}  
finally {  
    System.out.print(4);  
}  
}  
static void n() throws Exception {  
    try {  
        throw( new MyExc2() );  
    }  
    catch( MyExc1 t ) {  
    }  
    catch( Exception i ) {  
    }  
    catch( MyExc3 s ) {  
        throw( new MyExc3() );  
    }  
    finally {  
        System.out.print(5);  
    }  
}  
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

4. Dato un file contenente il seguente codice:

```
package pk;  
  
public class A {  
    public class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().

- D. Un'istanza della classe I può essere costruita con `new A().new I()`.  
E. Nessuna delle precedenti.
- 

5. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `t = w;`  
B. `t = z;`  
C. `t = (String []) w;`  
D. `w = (Object [] []) t;`  
E. `w = (Object [] []) z;`
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    String constructor;  
    int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c) {super(f,c);}  
  
    public void print() {  
        System.out.print(floors + " "  
                         + constructor + " ");  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

7. Quale output si ottiene invocando il metodo p?

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");  
        String s6;  
        String s8 = "";  
        s2 = s1.concat("abcd");  
        s6 = "";  
        a4 = a3;  
        q(s4, s6, s8, a4, a3);  
    }  
    void q(String s3, String s5, String s7,  
           float[] a1, Object a2){  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s5 == s7) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a2 == a1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}  
A. 100  
B. 011  
C. 111  
D. 000  
E. 101
```

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;
```

```

constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

9. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;

- D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
- B. YYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

12. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c){super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
                        + constructor + " ");  
    }  
}  
  
class Run {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class A1 {  
    public static void main(String[] argv)  
        throws Exception {  
        try {  
            p();  
            System.out.print(1);  
        }  
        catch( MyExc1 e ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 d ) {  
            System.out.print(3);  
            throw( new Exception() );  
        }  
        catch( Exception i ) {  
            System.out.print(4);  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
    static void p() throws Exception {  
        try {  
            System.out.print(6);  
            if (true) throw( new MyExc3() );  
            throw( new MyExc1() );  
        }  
        catch( MyExc1 t ) {  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 s ) {  
        }  
    }  
}
```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XXYY
  - B. YYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }

```

---

```

class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 17**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}
```

```
abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
    }
}
```

```

    }
    finally {
        System.out.print(5);
    }
}

static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1());
    }
    catch( MyExc1 t ) {
        throw( new MyExc1());
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual è il *sovraffaticcio* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

5. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

7. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
    }
}

```

```

        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                         + ((X) this).s
                         + ((Y) this).s
                         + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
- B. XYYY

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

9. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

10. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

11. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

13. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
```

```
Coffee() {
    System.out.println(brand);
}
Coffee(int ml){
    this.ml = ml;
    System.out.println(ml);
}
public static void main(String args[]) {
    Coffee ristretto = new Coffee();
    Coffee kazar = new Coffee();
    Coffee vanillio = new Coffee(40);
    ristretto = null;
    vanillio = ristretto;
    /* QUI */
    return;
}
A. 1
B. 2
C. 3
D. 4
E. 6
```

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
    }
}
```

```

    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {

```

```

        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 18**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}
```

```
}                                }

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

3. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
-

4. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

A. Jones Smith 50 Black

B. Jones 21 Smith 50 Black

C. Black Black 50 Black

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

A. Jones Smith 50 Black

B. Jones 21 Smith 50 Black

C. Black Black 50 Black

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

```

    s6 = "";
    a4 = a3;
    q(s4, s6, s8, a4, a3);
}
void q(String s3, String s5, String s7,
       float[] a1, Object a2){
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

#### 8. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

#### 9. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
    }
}

```

#### 10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {

```

```

        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

12. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

13. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {

```

```

try {
    throw( new MyExc2() );
}
catch( MyExc1 t ) {
}
catch( Exception i ) {
}
catch( MyExc3 s ) {
    throw( new MyExc3() );
}
finally {
    System.out.print(5);
}
}
}

```

- A. Errore a tempo di compilazione  
 B. 152  
 C. 1524  
 D. 154  
 E. Nessuna delle precedenti
- 
15. Qual è il *sovvraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A(){}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}

```

E. Il costruttore di una classe non può essere sovraccaricato.

---

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. 6
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 19

1. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

2. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

3. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends Error {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String[] argv){  
        try {  
            q();  
        }  
        catch( MyExc3 x ) {  
        }  
        catch( MyExc1 c ) {  
        }  
        catch( MyExc2 c ) {  
            System.out.print(1);  
        }  
        finally {  
            System.out.print(2);  
        }  
    }  
    static void q() {  
        try {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {
```

```

        String s = "X";
    }

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 6. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

#### 7. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {

```

```

        System.out.println(brand);
    }

    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }

    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

A. 1
B. 2
C. 3
D. 4
E. 6

```

---

#### 8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
    }
}

```

```

        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 152  
 C. 1524  
 D. 154  
 E. Nessuna delle precedenti
- 

9. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2) {
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100  
 B. 011  
 C. 111  
 D. 000  
 E. 101
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY  
 B. XYYY  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {

```

```

Building a = new Building(3,"Jones");
Building b = new Skyscraper(21,"Smith");
Skyscraper c = new Skyscraper(50,"Black");
a.print();
b.print();
c.print();
}

```

E. Errore a tempo di esecuzione.

---

- A. Jones Smith 50 Black  
 B. Jones 21 Smith 50 Black  
 C. Black Black 50 Black  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
 B. Jones 21 Smith 50 Black  
 C. Black Black 50 Black  
 D. Errore a tempo di compilazione.
- 

13. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.  
 B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
 C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().  
 D. Un'istanza della classe I può essere costruita con new A().new I().  
 E. Nessuna delle precedenti.
- 

14. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)  
 B. public byte f(Object y)  
 C. protected int f(String z)  
 D. int f(String s) throws Exception  
 E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}
}

```

```

public static void print() {
    System.out.print(floors + " "
        + constructor + " ");
}
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

A. Jones Smith 50 Black
B. Jones 21 Smith 50 Black
C. Black Black 50 Black
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        } catch( MyExc1 e ) {
            System.out.print(2);
        } catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        } catch( Exception i ) {
            System.out.print(4);
        } finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
            throw( new MyExc1() );
        } catch( MyExc1 t ) {
            throw( new MyExc1() );
        } catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 20**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

3. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

#### 4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
    }
}

```

```

        finally {
            System.out.print(5);
        }
    }
A. Errore a tempo di compilazione
B. 152
C. 1524
D. 154
E. Nessuna delle precedenti

```

---

#### 5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- 
7. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

- 
8. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
```

```
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

- 
9. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

- 
10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }
}
```

```

public void print() {
    System.out.print(constructor + " ");
}
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}
}

public void print() {
    System.out.print(floors + " "
        + constructor + " ");
}
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

11. Qual è il *sovraffaccimento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

```

```

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black

- B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

14. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

15. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

16. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 21**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
    }
}
```

```

}
catch( MyExc1 t ) {
}
catch( Exception i ) {
}
catch( MyExc3 s ) {
    throw( new MyExc3() );
}
finally {
    System.out.print(5);
}
}
}

```

- A. Errore a tempo di compilazione  
 B. 152  
 C. 1524  
 D. 154  
 E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black

- B. Jones 21 Smith 50 Black  
 C. Black Black 50 Black  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

5. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;  
 B. h = (C1) a;  
 C. h = (C1) t;  
 D. a = (Sub2) h;  
 E. Nessuno dei precedenti
- 

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
    }
}

```

```

        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34  
B. 342Exception in thread "main" MyExc3  
C. Errore a tempo di compilazione  
D. 342  
E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

9. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione  
B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array  
C. In Java tutti i tipi numerici sono con segno  
D. Un array possiede dei membri  
E. Un array vuoto può avere riferimento null
- 

10. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

---

11. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

---

12. Quale output si ottiene invocando il metodo p?

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");  
        String s6;  
        String s8 = "";  
        s2 = s1.concat("abcd");  
        s6 = "";  
        a4 = a3;  
        q(s4, s6, s8, a4, a3);  
    }  
    void q(String s3, String s5, String s7,  
          float[] a1, Object a2){  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s5 == s7) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a2 == a1) {  
            System.out.print(1);  
        }  
    }  
}
```

```
    } else {  
        System.out.print(0);  
    }  
}  
}  
A. 100  
B. 011  
C. 111  
D. 000  
E. 101
```

---

13. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

---

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
abstract class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
                        + ((X) this).s  
                        + ((Y) this).s  
                        + this.s);  
    }  
  
    public static void main(String[] args){  
        new Y().m();  
    }  
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    int ml = 20;  
    Coffee() {  
        System.out.println(brand);  
    }  
    Coffee(int ml) {  
        this.ml = ml;  
        System.out.println(ml);  
    }  
    public static void main(String args[]) {  
        Coffee ristretto = new Coffee();  
        Coffee kazar = new Coffee();  
        Coffee vanillio = new Coffee(40);  
        ristretto = null;  
        vanillio = ristretto;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
                        + ((X) this).s  
                        + ((Y) this).s  
                        + this.s);  
    }  
}  
  
public static void main(String[] args) {  
    new Y().m();  
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 22**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

3. Qual è il *sovraffaccimento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A() {}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.
```

---

4. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
    }
}

```

```

    }
    catch( Exception i ) {
        System.out.print(4);
    }
    finally {
        System.out.print(5);
    }
}
static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

#### 8. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
    }
}

```

```

        }
    finally {
        System.out.print(2);
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
}

```

- A. 34  
B. 342Exception in thread "main" MyExc3  
C. Errore a tempo di compilazione  
D. 342  
E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XXYY  
B. YYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

10. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().  
D. Un'istanza della classe I può essere costruita con new A().new I().  
E. Nessuna delle precedenti.
- 

11. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione  
B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array  
C. In Java tutti i tipi numerici sono con segno  
D. Un array possiede dei membri  
E. Un array vuoto può avere riferimento null
- 

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee(){
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1  
B. 2

C. 3

D. 4

E. 6

---

13. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        } catch( MyExc2 r ) {
        } catch( MyExc1 j ) {
        } catch( Exception e ) {
            System.out.print(3);
        } finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        } catch( MyExc1 t ) {
        } catch( Exception i ) {
        } catch( MyExc3 s ) {
            throw( new MyExc3() );
        } finally {
            System.out.print(5);
        }
    }
}
```

A. Errore a tempo di compilazione

B. 152

C. 1524

D. 154

E. Nessuna delle precedenti

---

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

A. Jones Smith 50 Black

B. Jones 21 Smith 50 Black

C. Black Black 50 Black

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

---

15. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
```

```

if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s5 == s7) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(a2 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}

A. 100
B. 011
C. 111
D. 000
E. 101

```

---

```

public static void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }
}

```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 23**

1. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}
```

```
class Run {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

3. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

4. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

5. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee(){
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
    }
}

```

```

        vanillio = ristretto;
        /* QUI */
        return;
    }
}

A. 1
B. 2
C. 3
D. 4
E. 6

```

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args){
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv){
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
    }
}

```

```

        catch( MyExc1 c ) {
    }
    catch( MyExc2 c ) {
        System.out.print(1);
    }
    finally {
        System.out.print(2);
    }
}
static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}

```

- A. 34  
B. 342Exception in thread "main" MyExc3  
C. Errore a tempo di compilazione  
D. 342  
E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
    }
}

```

```

        a.print();
        b.print();
        c.print();
    }
}

A. Jones Smith 50 Black
B. Jones 21 Smith 50 Black
C. Black Black 50 Black
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

A. XXYY  
B. XYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
    }
}

```

```

        catch( MyExc2 r ) {
    }
    catch( MyExc1 j ) {
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. Errore a tempo di compilazione  
B. 152  
C. 1524  
D. 154  
E. Nessuna delle precedenti
- 
11. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A(){}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.

```

---

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

```

```

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

13. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione  
B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array  
C. In Java tutti i tipi numerici sono con segno  
D. Un array possiede dei membri  
E. Un array vuoto può avere riferimento null
- 

14. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;

- B. t = z;  
C. t = (String []) w;  
D. w = (Object [] []) t;  
E. w = (Object [] []) z;
- 

15. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)  
B. public byte f(Object y)  
C. protected int f(String z)  
D. int f(String s) throws Exception  
E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class A1 {  
    public static void main(String[] argv)  
        throws Exception {  
        try {  
            p();  
            System.out.print(1);  
        }  
        catch( MyExc1 e ) {
```

```
            System.out.print(2);  
        }  
        catch( MyExc3 d ) {  
            System.out.print(3);  
            throw( new Exception() );  
        }  
        catch( Exception i ) {  
            System.out.print(4);  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
    static void p() throws Exception {  
        try {  
            System.out.print(6);  
            if (true) throw( new MyExc3() );  
            throw( new MyExc1() );  
        }  
        catch( MyExc1 t ) {  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 s ) {  
        }  
    }  
}
```

- 
- A. 635Exception in thread "main" java.lang.Exception  
B. 63Exception in thread "main" java.lang.Exception  
C. 615  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 24

1. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

2. Quale output si ottiene invocando il metodo p?

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");  
        String s6;  
        String s8 = "";  
        s2 = s1.concat("abcd");  
        s6 = "";  
        a4 = a3;  
        q(s4, s6, s8, a4, a3);  
    }  
    void q(String s3, String s5, String s7,  
          float[] a1, Object a2){  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s5 == s7) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a2 == a1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```
}  
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

3. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
                        + ((X) this).s  
                        + ((Y) this).s  
    }  
}
```

```

        + this.s);
    }

public static void main(String[] args) {
    new Y().m();
}
}

```

- A. XXYY  
B. XYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 
5. Qual è il *sovraffaticamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A(){}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.

```

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

A. Jones Smith 50 Black
B. Jones 21 Smith 50 Black
C. Black Black 50 Black
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
- B. XYYY

- C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
```

```

        this.ml = ml;
        System.out.println(ml);
    }
}

public static void main(String args[]) {
    Coffee ristretto = new Coffee();
    Coffee kazar = new Coffee();
    Coffee vanillio = new Coffee(40);
    ristretto = null;
    vanillio = ristretto;
    /* QUI */
    return;
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

## 12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

## 13. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

## 14. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

## 15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }
}

```

```

public static void print() {
    System.out.print(constructor + " ");
}
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 16. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }

```

```

class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 25**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
 B. Jones 21 Smith 50 Black  
 C. Black Black 50 Black  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

```

```

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
 B. Jones 21 Smith 50 Black  
 C. Black Black 50 Black  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY  
 B. XYYY  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.

---

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual'è il *sovraffunzione* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A() {}
C. private A(int i, String s) {}
```

- D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

8. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

9. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

10. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().

- D. Un'istanza della classe I può essere costruita con `new A().new I()`.  
E. Nessuna delle precedenti.
- 

- B. 011  
C. 111  
D. 000  
E. 101

11. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `t = w;`  
B. `t = z;`  
C. `t = (String []) w;`  
D. `w = (Object [] []) t;`  
E. `w = (Object [] []) z;`
- 

12. Quale output si ottiene invocando il metodo p?

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");  
        String s6;  
        String s8 = "";  
        s2 = s1.concat("abcd");  
        s6 = "";  
        a4 = a3;  
        q(s4, s6, s8, a4, a3);  
    }  
    void q(String s3, String s5, String s7,  
          float[] a1, Object a2){  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s5 == s7) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a2 == a1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

A. 100

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
                        + ((X) this).s  
                        + ((Y) this).s  
                        + this.s);  
    }  
}  
  
public static void main(String[] args){  
    new Y().m();  
}
```

- A. XXYY  
B. XYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    int ml = 20;  
    Coffee(){  
        System.out.println(brand);  
    }  
    Coffee(int ml){  
        this.ml = ml;  
        System.out.println(ml);  
    }  
    public static void main(String args[]){  
        Coffee ristretto = new Coffee();  
        Coffee kazar = new Coffee();  
        Coffee vanillio = new Coffee(40);  
    }  
}
```

```
    ristretto = null;
    vanillio = ristretto;
    /* QUI */
    return;
}
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
}
```

```
static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

16. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 26

1. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

A. 34

- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

3. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

4. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2) {
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black

- B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

```

```

public void print() {
    System.out.print(floors + " "
        + constructor + " ");
}
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

A. Jones Smith 50 Black
B. Jones 21 Smith 50 Black
C. Black Black 50 Black
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

A. 635Exception in thread "main" java.lang.Exception  
B. 63Exception in thread "main" java.lang.Exception  
C. 615  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

---

#### 8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
    }
}

```

```

        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception  
B. 63Exception in thread "main" java.lang.Exception  
C. 615  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 9. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().  
D. Un'istanza della classe I può essere costruita con new A().new I().  
E. Nessuna delle precedenti.
- 

#### 10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

```

static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 152  
 C. 1524  
 D. 154  
 E. Nessuna delle precedenti
- 

11. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione  
 B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array  
 C. In Java tutti i tipi numerici sono con segno  
 D. Un array possiede dei membri  
 E. Un array vuoto può avere riferimento null
- 

12. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)  
 B. public byte f(Object y)  
 C. protected int f(String z)  
 D. int f(String s) throws Exception  
 E. Nessuna delle precedenti.
- 

13. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;  
 B. h = (C1) a;  
 C. h = (C1) t;  
 D. a = (Sub2) h;  
 E. Nessuno dei precedenti
- 

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. 6
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

A. XXYY
B. XYYY
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;
}

```

```

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 27

1. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}
```

```
class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";
}

public void m() {
    System.out.print(super.s
        + ((X) this).s
        + ((Y) this).s
        + this.s);
}

public static void main(String[] args) {
    new Y().m();
}
}

A. XXYY
B. XYYY
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.
```

4. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
    }}
```

```

        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100  
 B. 011  
 C. 111  
 D. 000  
 E. 101

```

    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
 B. Jones 21 Smith 50 Black  
 C. Black Black 50 Black  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.

5. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione  
 B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array  
 C. In Java tutti i tipi numerici sono con segno  
 D. Un array possiede dei membri  
 E. Un array vuoto può avere riferimento null

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

```

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

9. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

A. 1

- B. 2
  - C. 3
  - D. 4
  - E. 6
- 

10. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

- ```
public class A {
    A(String s) {}
}
```
- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

A. Jones Smith 50 Black

- B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

13. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
    }
}
```

```

        }
    finally {
        System.out.print(2);
    }
}

static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
}

A. 34
B. 342Exception in thread "main" MyExc3
C. Errore a tempo di compilazione
D. 342
E. Nessuna delle precedenti

```

---

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception  
B. 63Exception in thread "main" java.lang.Exception  
C. 615  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 28**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

2. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

A. XXYY
```

B. XYY

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
4. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    int ml = 20;  
    Coffee() {  
        System.out.println(brand);  
    }  
    Coffee(int ml) {  
        this.ml = ml;  
        System.out.println(ml);  
    }  
    public static void main(String args[]) {  
        Coffee ristretto = new Coffee();  
        Coffee kazar = new Coffee();  
        Coffee vanillio = new Coffee(40);  
        ristretto = null;  
        vanillio = ristretto;  
        /* QUI */  
        return;  
    }  
}
```

A. 1

B. 2

C. 3

D. 4

E. 6

- 
5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c) {super(f, c);}  
}
```

```
    public static void print() {  
        System.out.print(floors + " "  
                         + constructor + " ");  
    }  
}  
  
class Run {  
    public static void main(String[] args) {  
        Building a = new Building(3, "Jones");  
        Building b = new Skyscraper(21, "Smith");  
        Skyscraper c = new Skyscraper(50, "Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

A. Jones Smith 50 Black

B. Jones 21 Smith 50 Black

C. Black Black 50 Black

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

- 
6. Qual'è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends Error {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String[] argv) {  
        try {  
            q();  
        }  
        catch( MyExc3 x ) {  
        }  
        catch( MyExc1 c ) {  
        }  
        catch( MyExc2 c ) {  
            System.out.print(1);  
        }  
        finally {  
            System.out.print(2);  
        }  
    }  
    static void q() {  
        try {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc3() );  
        }  
    }  
}
```

A. 34

B. 342Exception in thread "main" MyExc3

C. Errore a tempo di compilazione

D. 342

E. Nessuna delle precedenti

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(){}  
    Skyscraper(int f, String c) {super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
            + constructor + " ");  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

A. Jones Smith 50 Black

B. Jones 21 Smith 50 Black

C. Black Black 50 Black

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

---

8. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

A. public void f(String x)

B. public byte f(Object y)

C. protected int f(String z)

D. int f(String s) throws Exception

E. Nessuna delle precedenti.

---

9. Quale output si ottiene invocando il metodo p?

```
class B {  
    private String s1 = "";  
    private String s2;  
    private float [] a3 = new float [8];  
    private float [] a4;  
    void p() {  
        String s4 = s1.concat("abcd");  
        String s6;  
        String s8 = "";  
        s2 = s1.concat("abcd");  
        s6 = "";  
        a4 = a3;  
        q(s4, s6, s8, a4, a3);  
    }  
    void q(String s3, String s5, String s7,  
           float[] a1, Object a2){  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s5 == s7) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a2 == a1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

A. 100

B. 011

C. 111

D. 000

E. 101

---

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    String constructor;  
    int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;
```

```

}

public void print() {
    System.out.print(constructor + " ");
}
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

A. Jones Smith 50 Black
B. Jones 21 Smith 50 Black
C. Black Black 50 Black
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

}

static void p() throws Exception {
    try {
        System.out.print(6);
        if (true) throw( new MyExc3());
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}

A. 635Exception in thread "main" java.lang.Exception
B. 63Exception in thread "main" java.lang.Exception
C. 615
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

12. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

---

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
}

```

13. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

14. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```
Sub2 a;  
C1 h;  
A t;  
t = new C1();  
a = new Sub2();  
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

15. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
            + ((X) this).s  
            + ((Y) this).s  
            + this.s);  
    }  
  
    public static void main(String[] args) {  
        new Y().m();  
    }  
}  
  
A. XXYY  
B. XYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 29**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
```

```
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper() {}
    Skyscraper(int f, String c) {super(f, c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3, "Jones");
        Building b = new Skyscraper(21, "Smith");
        Skyscraper c = new Skyscraper(50, "Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

3. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 152  
C. 1524  
D. 154  
E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

```

```

public void m() {
    System.out.print(super.s
        + ((X) this).s
        + ((Y) this).s
        + this.s);
}

public static void main(String[] args) {
    new Y().m();
}

A. XXYY
B. XYYY
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

6. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)  
B. public byte f(Object y)  
C. protected int f(String z)  
D. int f(String s) throws Exception  
E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

9. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        } catch( MyExc1 e ) {
            System.out.print(2);
        } catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        } catch( Exception i ) {
            System.out.print(4);
        } finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
            throw( new MyExc1() );
        } catch( MyExc1 t ) {
            throw( new MyExc1() );
        } catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34  
B. 342Exception in thread "main" MyExc3  
C. Errore a tempo di compilazione  
D. 342  
E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {

```

```

        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

13. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6
- 

14. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011
- C. 111
- D. 000

E. 101

---

15. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A() {}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.

```

---

16. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 30**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv){
        try {
            q();
        } catch( MyExc3 x ) {
        } catch( MyExc1 c ) {
        } catch( MyExc2 c ) {
            System.out.print(1);
        } finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        } finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

3. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
```

```

String s8 = "";
s2 = s1.concat("abcd");
s6 = "";
a4 = a3;
q(s4, s6, s8, a4, a3);
}
void q(String s3, String s5, String s7,
       float[] a1, Object a2){
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 100  
 B. 011  
 C. 111  
 D. 000  
 E. 101
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

A. Jones Smith 50 Black
B. Jones 21 Smith 50 Black
C. Black Black 50 Black
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

A. XXYY  
 B. XYYY  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

```

```

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

8. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;

- B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

9. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

10. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {

```

```

}
catch( Exception e ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
}
}

static void n() throws Exception {
try {
    throw( new MyExc2() );
}
catch( MyExc1 t ) {
}
catch( Exception i ) {
}
catch( MyExc3 s ) {
    throw( new MyExc3() );
}
finally {
    System.out.print(5);
}
}
}

```

- A. Errore a tempo di compilazione  
B. 152  
C. 1524  
D. 154  
E. Nessuna delle precedenti
- 
12. Qual è il *sovvraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A(){}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.

```

---

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
    try {
        p();
    }
}

```

```

        System.out.print(1);
    }
    catch( MyExc1 e ) {
        System.out.print(2);
    }
    catch( MyExc3 d ) {
        System.out.print(3);
        throw( new Exception() );
    }
    catch( Exception i ) {
        System.out.print(4);
    }
    finally {
        System.out.print(5);
    }
}
static void p() throws Exception {
try {
    System.out.print(6);
    if (true) throw( new MyExc3());
    throw( new MyExc1() );
}
catch( MyExc1 t ) {
    throw( new MyExc1() );
}
catch( MyExc2 s ) {
}
}

```

- A. 635Exception in thread "main" java.lang.Exception  
B. 63Exception in thread "main" java.lang.Exception  
C. 615  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee(){
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c){super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
                        + constructor + " ");  
    }  
}
```

```
class Run {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 31**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
```

```
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
```

```

}
catch( Exception e ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
}
}
static void n() throws Exception {
try {
    throw( new MyExc2() );
}
catch( MyExc1 t ) {
}
catch( Exception i ) {
}
catch( MyExc3 s ) {
    throw( new MyExc3() );
}
finally {
    System.out.print(5);
}
}
}

```

- A. Errore a tempo di compilazione  
B. 152  
C. 1524  
D. 154  
E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34  
B. 342Exception in thread "main" MyExc3  
C. Errore a tempo di compilazione  
D. 342  
E. Nessuna delle precedenti
-

6. Dire quale delle seguenti affermazioni è falsa:
- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 
7. Qual'è l'output di questo codice?
- ```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] args)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```
- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 
8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?
- ```
public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```
- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 
9. Dato un file contenente il seguente codice:
- ```
package pk;

public class A {
    public class I {
    }
}
```
- Dire quale delle seguenti affermazioni è vera:
- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 
10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Qual è il *sovraff장imento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A(){}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.

```

---

12. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception

E. Nessuna delle precedenti.

---

13. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

14. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

15. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
}

```

```

}

void q(String s3, String s5, String s7,
      float[] a1, Object a2){
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

A. 100
B. 011
C. 111
D. 000
E. 101

```

---

```

floors = f;
constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {

```

---

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 32

1. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
                        + ((X) this).s  
                        + ((Y) this).s  
                        + this.s);  
    }  
  
    public static void main(String[] args){  
        new Y().m();  
    }  
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

- E. Nessuna delle precedenti.

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    int ml = 20;  
    Coffee(){  
        System.out.println(brand);  
    }  
    Coffee(int ml){  
        this.ml = ml;  
        System.out.println(ml);  
    }  
    public static void main(String args[]){  
        Coffee ristretto = new Coffee();  
        Coffee kazar = new Coffee();  
        Coffee vanillio = new Coffee(40);  
        ristretto = null;  
        vanillio = ristretto;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    String constructor;  
    int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public void print() {  
        System.out.print(constructor + " ");  
    }  
}
```

```

}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
        }
    }
}

```

```

        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;

- B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

9. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
-

11. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
- D. Un'istanza della classe I può essere costruita con new A().new I().
- E. Nessuna delle precedenti.

---

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
- B. YYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

14. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
```

```

float[] a1, Object a2){
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s5 == s7) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(a2 == a1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

15. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

---

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

16. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 33**

1. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new A.I()`.
  - D. Un'istanza della classe I può essere costruita con `new A().new I()`.
  - E. Nessuna delle precedenti.
- 

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
```

```
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

3. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

4. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.

9. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
}
```

A. XXYY  
B. XYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011

C. 111

D. 000

E. 101

---

13; Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black

C. Black Black 50 Black

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

- 
15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    int ml = 20;  
    Coffee(){  
        System.out.println(brand);  
    }  
    Coffee(int ml){  
        this.ml = ml;  
        System.out.println(ml);  
    }  
    public static void main(String args[]){  
        Coffee ristretto = new Coffee();  
        Coffee kazar = new Coffee();  
        Coffee vanillio = new Coffee(40);  
        ristretto = null;  
        vanillio = ristretto;  
        /* QUI */  
        return;  
    }  
}
```

A. 1

B. 2

C. 3

D. 4

E. 6

- 
16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
abstract class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
            + ((X) this).s  
            + ((Y) this).s  
            + this.s);  
    }  
  
    public static void main(String[] args) {  
        new Y().m();  
    }  
}
```

A. XXYY

B. XYYY

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 34**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

2. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new A.I()`.
  - D. Un'istanza della classe I può essere costruita con `new A().new I()`.
  - E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black

C. Black Black 50 Black

- D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XXYY
  - B. YYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 8. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

---

#### 9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
- B. 63Exception in thread "main" java.lang.Exception
- C. 615
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

10. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

11. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
  - B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
  - C. In Java tutti i tipi numerici sono con segno
  - D. Un array possiede dei membri
  - E. Un array vuoto può avere riferimento null
- 

12. Qual è il *sovraffunzione* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

13. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black"); }
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black
- 

- C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee(){
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

A. 1
B. 2
C. 3
D. 4
E. 6

```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 35**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
  - B. 342Exception in thread "main" MyExc3
  - C. Errore a tempo di compilazione
  - D. 342
  - E. Nessuna delle precedenti
- 

2. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new A.I()`.
  - D. Un'istanza della classe I può essere costruita con `new A().new I()`.
  - E. Nessuna delle precedenti.
- 

3. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

---

4. Date le dichiarazioni:

```
String [] t;
Object [] [] w;
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
- B. t = z;
- C. t = (String []) w;
- D. w = (Object [] []) t;
- E. w = (Object [] []) z;

---

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 152
- C. 1524
- D. 154
- E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
- B. h = (C1) a;
- C. h = (C1) t;
- D. a = (Sub2) h;
- E. Nessuno dei precedenti

---

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3() );
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
```

```
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";
```

```

public void m() {
    System.out.print(super.s
        + ((X) this).s
        + ((Y) this).s
        + this.s);
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black

- C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Qual è il *sovraffondamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A() {}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.

```

---

13. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- B. 2
- C. 3
- D. 4
- E. 6

15. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}
```

```
    }

    class Skyscraper extends Building {
        Skyscraper() {}
        Skyscraper(int f, String c) {super(f,c);}

        public static void print() {
            System.out.print(floors + " "
                + constructor + " ");
        }
    }

    class Main {
        public static void main(String[] args) {
            Building a = new Building(3,"Jones");
            Building b = new Skyscraper(21,"Smith");
            Skyscraper c = new Skyscraper(50,"Black");
            a.print();
            b.print();
            c.print();
        }
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 36**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Quale output si ottiene invocando il metodo p?

```
class B {
```

```
private String s1 = "";
private String s2;
private float [] a3 = new float [8];
private float [] a4;
void p() {
    String s4 = s1.concat("abcd");
    String s6;
    String s8 = "";
    s2 = s1.concat("abcd");
    s6 = "";
    a4 = a3;
    q(s4, s6, s8, a4, a3);
}
void q(String s3, String s5, String s7,
       float[] a1, Object a2){
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 100
- B. 011
- C. 111
- D. 000
- E. 101

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
```

```

try {
    q();
}
catch( MyExc3 x ) {
}
catch( MyExc1 c ) {
}
catch( MyExc2 c ) {
    System.out.print(1);
}
finally {
    System.out.print(2);
}
}

static void q() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
}

```

- A. 34  
B. 342Exception in thread "main" MyExc3  
C. Errore a tempo di compilazione  
D. 342  
E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XXYY  
B. XYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

6. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

A. public A(String b) {}

```

- B. void A() {}  
C. private A(int i, String s) {}  
D. public Object A(int i, String s) {}  
E. Il costruttore di una classe non può essere sovraccaricato.
- 

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione  
B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array  
C. In Java tutti i tipi numerici sono con segno  
D. Un array possiede dei membri  
E. Un array vuoto può avere riferimento null
- 

8. Date le dichiarazioni:

```
class C1 extends A {...}  
class A extends Object {...}  
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;  
C1 h;  
A t;  
t = new C1();  
a = new Sub2();  
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;  
B. h = (C1) a;  
C. h = (C1) t;  
D. a = (Sub2) h;  
E. Nessuno dei precedenti
- 

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class A1 {  
    public static void main(String[] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            n();  
            System.out.print(2);  
        }  
    }
```

```
        catch( MyExc2 r ) {}  
        catch( MyExc1 j ) {}  
        catch( Exception e ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            throw( new MyExc2() );  
        }  
        catch( MyExc1 t ) {}  
        catch( Exception i ) {}  
        catch( MyExc3 s ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione  
B. 152  
C. 1524  
D. 154  
E. Nessuna delle precedenti
- 

10. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;  
B. t = z;  
C. t = (String []) w;  
D. w = (Object [] []) t;  
E. w = (Object [] []) z;
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY  
B. YYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

## 12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {

```

```

            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}
```

- A. 635Exception in thread "main" java.lang.Exception  
B. 63Exception in thread "main" java.lang.Exception  
C. 615  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

## 13. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)  
B. public byte f(Object y)  
C. protected int f(String z)  
D. int f(String s) throws Exception  
E. Nessuna delle precedenti.
- 

## 14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}
```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

```

ristretto = null;
vanillio = ristretto;
/* QUI */
return;
A. 1
B. 2
C. 3
D. 4
E. 6

```

---

- A. Jones Smith 50 Black  
 B. Jones 21 Smith 50 Black  
 C. Black Black 50 Black  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.

15. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]) {
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
    }
}

```

16. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.  
 B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
 C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().  
 D. Un'istanza della classe I può essere costruita con new A().new I().  
 E. Nessuna delle precedenti.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 37**

1. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
```

```
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
                    throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {

```

```

Skyscraper(int f, String c){super(f,c);}

public static void print() {
    System.out.print(floors + " "
                    + constructor + " ");
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5; Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

B. 342Exception in thread "main" MyExc3

C. Errore a tempo di compilazione

D. 342

E. Nessuna delle precedenti

---

6. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    int ml = 20;  
    Coffee() {  
        System.out.println(brand);  
    }  
    Coffee(int ml) {  
        this.ml = ml;  
        System.out.println(ml);  
    }  
    public static void main(String args[]) {  
        Coffee ristretto = new Coffee();  
        Coffee kazar = new Coffee();  
        Coffee vanillio = new Coffee(40);  
        ristretto = null;  
        vanillio = ristretto;  
        /* QUI */  
        return;  
    }  
}
```

A. 1

B. 2

C. 3

D. 4

E. 6

---

7. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

A. t = w;

B. t = z;

C. t = (String []) w;

D. w = (Object [] []) t;

E. w = (Object [] []) z;

---

8. Qual è il *sovraffaccimento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
```

```
    A(String s) {}
```

```
}
```

A. public A(String b) {}

B. void A() {}

C. private A(int i, String s) {}

D. public Object A(int i, String s) {}

E. Il costruttore di una classe non può essere sovraccaricato.

---

9. Date le dichiarazioni:

```
class C1 extends A {...}  
class A extends Object {...}  
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;  
C1 h;  
A t;  
t = new C1();  
a = new Sub2();  
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

A. a = (Sub2) t;

B. h = (C1) a;

C. h = (C1) t;

D. a = (Sub2) h;

E. Nessuno dei precedenti

---

10. Dato un file contenente il seguente codice:

```
package pk;  
  
public class A {  
    public class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

A. Un'istanza della classe I può essere costruita solo all'interno della classe A.

B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().

D. Un'istanza della classe I può essere costruita con new A().new I().

E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(){  
    }  
    Skyscraper(int f, String c) {super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
                         + constructor + " ");  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

- 
12. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
                         + ((X) this).s  
                         + ((Y) this).s  
                         + this.s);  
    }  
}  
  
public static void main(String[] args) {  
    new Y().m();  
}
```

A. XXYY  
B. XYYY  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {  
    void m();  
}  
  
abstract class X implements I {  
    String s = "X";  
}  
  
class Y extends X {  
    String s = "Y";  
  
    public void m() {  
        System.out.print(super.s  
                         + ((X) this).s  
                         + ((Y) this).s  
                         + this.s);  
    }  
}  
  
public static void main(String[] args) {  
    new Y().m();  
}
```

A. XXYY

- B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
```

```
        System.out.print(6);
        if (true) throw( new MyExc3() );
        throw( new MyExc1() );
    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
```

- 
- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 38**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";
```

```
public void m() {
    System.out.print(super.s
        + ((X) this).s
        + ((Y) this).s
        + this.s);
}

public static void main(String[] args) {
    new Y().m();
}
```

- A. XXYY
- B. XYYY
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4

E. 6

---

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}
```

- A. 34
- B. 342Exception in thread "main" MyExc3
- C. Errore a tempo di compilazione
- D. 342
- E. Nessuna delle precedenti

---

5. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public static void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
- B. Jones 21 Smith 50 Black
- C. Black Black 50 Black
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

7. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
}
```

```

void q(String s3, String s5, String s7,
      float[] a1, Object a2){
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s5 == s7) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a2 == a1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 100  
 B. 011  
 C. 111  
 D. 000  
 E. 101
- 

**8. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
    }
}

```

```

        b.print();
        c.print();
    }
}

A. Jones Smith 50 Black
B. Jones 21 Smith 50 Black
C. Black Black 50 Black
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

**9. Date le dichiarazioni:**

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;  
 B. t = z;  
 C. t = (String []) w;  
 D. w = (Object [] []) t;  
 E. w = (Object [] []) z;
- 

**10. Qual'è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
    }
}

```

```

    }
    catch( MyExc1 t ) {
        throw( new MyExc1() );
    }
    catch( MyExc2 s ) {
    }
}
}

```

- A. 635Exception in thread "main" java.lang.Exception  
B. 63Exception in thread "main" java.lang.Exception  
C. 615  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

11. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new A.I()`.  
D. Un'istanza della classe I può essere costruita con `new A().new I()`.  
E. Nessuna delle precedenti.
- 

12. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. `public void f(String x)`  
B. `public byte f(Object y)`  
C. `protected int f(String z)`  
D. `int f(String s) throws Exception`  
E. Nessuna delle precedenti.
- 

13. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A() {}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.

```

---

14. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `a = (Sub2) t;`  
B. `h = (C1) a;`  
C. `h = (C1) t;`  
D. `a = (Sub2) h;`  
E. Nessuno dei precedenti
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
    finally {
        System.out.print(4);
    }
}

```

```

}

static void n() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( MyExc1 t ) {
    }
    catch( Exception i ) {
    }
    catch( MyExc3 s ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(5);
    }
}

A. Errore a tempo di compilazione
B. 152
C. 1524
D. 154
E. Nessuna delle precedenti

```

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;

    Building(int f, String c) {
        floors = f;
    }
}

```

```

    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(int f, String c){super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Run {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

A. Jones Smith 50 Black
B. Jones 21 Smith 50 Black
C. Black Black 50 Black
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 39**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            System.out.print(1);
            n();
            System.out.print(2);
        }
        catch( MyExc2 r ) {
        }
        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 152
  - C. 1524
  - D. 154
  - E. Nessuna delle precedenti
- 

2. Dato un file contenente il seguente codice:

```
package pk;

public class A {
    public class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con new A.I().
  - D. Un'istanza della classe I può essere costruita con new A().new I().
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
    String constructor;
    int floors;

    Building(int f, String c) {
        floors = f;
        constructor = c;
    }

    public void print() {
        System.out.print(constructor + " ");
    }
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}
```

```

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Quale output si ottiene invocando il metodo p?

```

class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 100
- B. 011
- C. 111
- D. 000

E. 101

---

5. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand = "Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml) {
        this.ml = ml;
        System.out.println(ml);
    }
}
public static void main(String args[]) {
    Coffee ristretto = new Coffee();
    Coffee kazar = new Coffee();
    Coffee vanillio = new Coffee(40);
    ristretto = null;
    vanillio = ristretto;
    /* QUI */
    return;
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
                        + ((X) this).s
                        + ((Y) this).s
                        + this.s);
    }
}

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
  - B. YYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c){super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
                        + constructor + " ");  
    }  
}  
  
class Run {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Date le dichiarazioni:

```
class C1 extends A {...}  
class A extends Object {...}  
class Sub2 extends A {...}
```

e le inizializzazioni di variabile:

```
Sub2 a;  
C1 h;  
A t;  
t = new C1();  
a = new Sub2();  
h = new C1();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
- 

9. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

10. Date le dichiarazioni:

```
String [] t;  
Object [] [] w;  
Object [] z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception  
B. 63Exception in thread "main" java.lang.Exception  
C. 615  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

12. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione  
B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array  
C. In Java tutti i tipi numerici sono con segno  
D. Un array possiede dei membri  
E. Un array vuoto può avere riferimento null
- 

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String[] argv) {
        try {
            q();
        }
        catch( MyExc3 x ) {
        }
        catch( MyExc1 c ) {
        }
        catch( MyExc2 c ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void q() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

- A. 34  
B. 342Exception in thread "main" MyExc3  
C. Errore a tempo di compilazione  
D. 342  
E. Nessuna delle precedenti
- 

14. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)  
B. public byte f(Object y)  
C. protected int f(String z)  
D. int f(String s) throws Exception  
E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args){
        new Y().m();
    }
}

A. XXYY
B. XYXY
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

```

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    static String constructor;
    static int floors;
}

```

---

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 40**

1. Quale output si ottiene invocando il metodo p?

```
class B {
    private String s1 = "";
    private String s2;
    private float [] a3 = new float [8];
    private float [] a4;
    void p() {
        String s4 = s1.concat("abcd");
        String s6;
        String s8 = "";
        s2 = s1.concat("abcd");
        s6 = "";
        a4 = a3;
        q(s4, s6, s8, a4, a3);
    }
    void q(String s3, String s5, String s7,
           float[] a1, Object a2){
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s5 == s7) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a2 == a1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 100
  - B. 011
  - C. 111
  - D. 000
  - E. 101
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {
```

```
static String constructor;
static int floors;

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public static void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(){}
    Skyscraper(int f, String c) {super(f,c);}

    public static void print() {
        System.out.print(floors + " "
                        + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}
```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
interface I {
    void m();
```

```

}

abstract class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }

    public static void main(String[] args) {
        new Y().m();
    }
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Dato un file contenente il seguente codice:

```

package pk;

public class A {
    public class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe A.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'interno del pacchetto pk, un'istanza della classe I può essere costruita con `new A.I()`.
  - D. Un'istanza della classe I può essere costruita con `new A().new I()`.
  - E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Building {
    String constructor;
    int floors;
}

```

```

Building(int f, String c) {
    floors = f;
    constructor = c;
}

public void print() {
    System.out.print(constructor + " ");
}

class Skyscraper extends Building {
    Skyscraper(int f, String c) {super(f,c);}

    public void print() {
        System.out.print(floors + " "
            + constructor + " ");
    }
}

class Main {
    public static void main(String[] args) {
        Building a = new Building(3,"Jones");
        Building b = new Skyscraper(21,"Smith");
        Skyscraper c = new Skyscraper(50,"Black");
        a.print();
        b.print();
        c.print();
    }
}

```

- A. Jones Smith 50 Black
  - B. Jones 21 Smith 50 Black
  - C. Black Black 50 Black
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

interface I {
    void m();
}

class X implements I {
    String s = "X";
}

class Y extends X {
    String s = "Y";

    public void m() {
        System.out.print(super.s
            + ((X) this).s
            + ((Y) this).s
            + this.s);
    }
}

```

```

public static void main(String[] args) {
    new Y().m();
}

```

- A. XXYY
  - B. XYYY
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Date le dichiarazioni:

```

String [] t;
Object [] [] w;
Object [] z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. t = w;
  - B. t = z;
  - C. t = (String []) w;
  - D. w = (Object [] []) t;
  - E. w = (Object [] []) z;
- 

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    int ml = 20;
    Coffee() {
        System.out.println(brand);
    }
    Coffee(int ml){
        this.ml = ml;
        System.out.println(ml);
    }
    public static void main(String args[]){
        Coffee ristretto = new Coffee();
        Coffee kazar = new Coffee();
        Coffee vanillio = new Coffee(40);
        ristretto = null;
        vanillio = ristretto;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
- 

E. 6

---

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class A1 {
    public static void main(String[] argv)
        throws Exception {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc1 e ) {
            System.out.print(2);
        }
        catch( MyExc3 d ) {
            System.out.print(3);
            throw( new Exception() );
        }
        catch( Exception i ) {
            System.out.print(4);
        }
        finally {
            System.out.print(5);
        }
    }
    static void p() throws Exception {
        try {
            System.out.print(6);
            if (true) throw( new MyExc3());
            throw( new MyExc1() );
        }
        catch( MyExc1 t ) {
            throw( new MyExc1() );
        }
        catch( MyExc2 s ) {
        }
    }
}

```

- A. 635Exception in thread "main" java.lang.Exception
  - B. 63Exception in thread "main" java.lang.Exception
  - C. 615
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Dire quale delle seguenti affermazioni è falsa:

- A. Un oggetto può non esistere dopo la sua dichiarazione
- B. La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C. In Java tutti i tipi numerici sono con segno
- D. Un array possiede dei membri
- E. Un array vuoto può avere riferimento null

- 
11. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}  
  
A. public A(String b) {}  
B. void A() {}  
C. private A(int i, String s) {}  
D. public Object A(int i, String s) {}  
E. Il costruttore di una classe non può essere sovraccaricato.
```

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends Error {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String[] argv) {  
        try {  
            q();  
        }  
        catch( MyExc3 x ) {  
        }  
        catch( MyExc1 c ) {  
        }  
        catch( MyExc2 c ) {  
            System.out.print(1);  
        }  
        finally {  
            System.out.print(2);  
        }  
    }  
    static void q() {  
        try {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 34  
B. 342Exception in thread "main" MyExc3  
C. Errore a tempo di compilazione  
D. 342  
E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Building {  
    static String constructor;  
    static int floors;  
  
    Building(int f, String c) {  
        floors = f;  
        constructor = c;  
    }  
  
    public static void print() {  
        System.out.print(constructor + " ");  
    }  
}  
  
class Skyscraper extends Building {  
    Skyscraper(int f, String c){super(f,c);}  
  
    public static void print() {  
        System.out.print(floors + " "  
                         + constructor + " ");  
    }  
}  
  
class Run {  
    public static void main(String[] args) {  
        Building a = new Building(3,"Jones");  
        Building b = new Skyscraper(21,"Smith");  
        Skyscraper c = new Skyscraper(50,"Black");  
        a.print();  
        b.print();  
        c.print();  
    }  
}
```

- A. Jones Smith 50 Black  
B. Jones 21 Smith 50 Black  
C. Black Black 50 Black  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class A1 {  
    public static void main(String[] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            n();  
            System.out.print(2);  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
}
```

```

        catch( MyExc1 j ) {
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
    static void n() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( MyExc1 t ) {
        }
        catch( Exception i ) {
        }
        catch( MyExc3 s ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 152  
 C. 1524  
 D. 154  
 E. Nessuna delle precedenti
- 

15. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

---

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

16. Date le dichiarazioni:

```

class C1 extends A {...}
class A extends Object {...}
class Sub2 extends A {...}

```

e le inizializzazioni di variabile:

```

Sub2 a;
C1 h;
A t;
t = new C1();
a = new Sub2();
h = new C1();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. a = (Sub2) t;
  - B. h = (C1) a;
  - C. h = (C1) t;
  - D. a = (Sub2) h;
  - E. Nessuno dei precedenti
-

**Prova n. 1**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 2**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 3**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 4**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 5**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 6**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 8**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 9**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 10**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 11**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 12**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 13**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 15**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 16**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

February 17, 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 1:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 2:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 3:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 4:

1		■					
2				■			
3			■				
4						■	
5			■				
6				■			
7			■				
8					■		
9				■			
10				■			
11					■		
12				■			
13						■	
14					■		
15				■			
16	■						

Risultato prova n. 5:

1			■				
2				■			
3						■	
4	■						
5				■			
6						■	
7					■		
8				■			
9					■		
10			■				
11	■						
12				■			
13					■		
14				■			
15			■				
16				■			

Risultato prova n. 6:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 7:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 8:

1					
2			■		
3				■	
4					■
5				■	
6		■			
7			■		
8	■				
9			■		
10	■				
11			■		
12			■		
13				■	
14			■		
15					■
16				■	

Risultato prova n. 9:

## Risultato prova n. 10:

1						
2			■			
3				■		
4				■		
5				■		
6				■		
7						■
8					■	
9				■		
10	■					
11			■			
12				■		
13	■					
14						■
15					■	
16		■				

Risultato prova n. 11:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 12:

1		■					
2					■	■	
3				■	■		
4				■	■		
5				■	■		
6		■					
7						■	■
8					■	■	
9	■	■					
10						■	■
11				■	■		
12					■	■	
13		■	■				
14				■	■		
15				■	■		
16		■	■				

Risultato prova n. 13:

1		■					
2			■				
3				■	■		
4					■	■	
5					■	■	
6			■				
7						■	■
8						■	■
9							■
10			■				
11				■			
12							■
13					■		■
14	■						
15				■			
16				■			

Risultato prova n. 14:

A 16x16 grid with red squares at the following coordinates:

- (1, 1)
- (2, 2)
- (3, 3)
- (4, 4)
- (5, 5)
- (6, 6)
- (7, 7)
- (8, 8)
- (9, 9)
- (10, 10)
- (11, 11)
- (12, 12)
- (13, 13)
- (14, 14)
- (15, 15)
- (16, 16)

## Risultato prova n. 15:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 16:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 17:

1		■					
2					■	■	
3						■	■
4				■	■		
5				■	■		
6		■					
7			■	■			
8				■	■		
9		■					
10			■	■			
11					■	■	
12					■	■	
13						■	■
14	■	■					
15			■	■			
16		■					

Risultato prova n. 18:

1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																

Risultato prova n. 19:

1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Risultato prova n. 20:

1					
2					
3		■			
4				■	
5				■	
6					■
7			■		
8					■
9					■
10				■	
11				■	
12			■		
13					■
14		■			
15			■		
16				■	

Risultato prova n. 21:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 22:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 23:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 24:

1		■					
2				■	■		
3						■	■
4			■				
5		■					
6				■	■		
7				■	■		
8				■	■		
9						■	■
10					■	■	
11						■	■
12			■	■			
13				■	■		
14			■	■			
15					■	■	
16				■	■		

Risultato prova n. 25:

## Risultato prova n. 26:

1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	

Risultato prova n. 27:

1		■					
2					■	■	
3		■					
4			■				
5				■			
6					■	■	
7					■	■	
8				■			
9			■	■			
10			■	■			
11				■			
12						■	■
13						■	■
14				■	■		
15				■	■		
16				■			

Risultato prova n. 28:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 29:

1						
2						
3			■	■		
4			■	■		
5	■	■				
6				■	■	
7					■	■
8						■
9						■
10				■	■	
11	■	■				
12				■	■	
13				■	■	
14			■	■		
15				■	■	
16				■		

Risultato prova n. 30:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 31:

1						
2						
3			■	■		
4			■	■		
5				■	■	
6				■	■	
7	■	■				
8					■	■
9					■	■
10				■	■	
11				■	■	
12	■	■				
13				■	■	
14			■	■		
15				■	■	
16				■		

Risultato prova n. 32:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 33:

1			■				
2				■			
3	■						
4					■		
5				■			
6	■						
7				■			
8		■					
9				■			
10						■	
11						■	
12				■			
13				■			
14					■		
15					■		
16		■					

Risultato prova n. 34:

1					
2					
3			■		
4					■
5				■	
6	■				
7				■	
8				■	
9	■				
10				■	
11		■			
12				■	
13				■	
14		■			
15					■
16				■	

Risultato prova n. 35:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 36:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 37:

1		■					
2				■			
3			■				
4					■		
5						■	
6					■		
7			■				
8			■				
9						■	
10				■			
11					■		
12				■			
13				■			
14				■			
15	■						
16				■			

Risultato prova n. 38:

1		■					
2					■	■	
3			■	■			
4			■	■			
5			■	■			
6	■	■					
7				■	■		
8				■	■		
9				■	■		
10						■	■
11				■	■		
12						■	■
13					■	■	
14				■	■		
15				■	■		
16					■	■	

Risultato prova n. 39:

1			■		
2				■	
3	■				
4				■	
5		■			
6			■		
7				■	
8		■			
9			■		
10				■	
11			■		
12				■	
13			■		
14	■				
15			■		
16			■		

Risultato prova n. 40:



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 1**

1. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
```

```
String s = "B";
public void m() {
    System.out.print(
        super.s + s);
}
public void g() {
    ((A)this).g();
}
public static void main(String[] args) {
    new B().m();
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)

- B. public String f(byte k)  
**C. protected String f(int k)**  
 D. String f(int i) throws Exception  
 E. Nessuna delle precedenti.
- 

5. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
}
public static void main (String[] args){
    new B().f();
}
```

- A. o1  
 B. o2  
 C. o3  
 D. o1 e o2  
**E. Nessuna delle precedenti.**
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
```

```
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 341**  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
}
public static void main(String[] args) {
    new B().m();
}
```

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

8. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
  
class Palm extends Tree {  
    public static void main(String argv[]){  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc2 g ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc3() );  
        }
```

```
}  
catch( MyExc3 c ) {  
    throw( new MyExc3() );  
}  
finally {  
    System.out.print(4);  
}  
}  
}
```

A. 14Exception in thread main MyExc3

B. Errore a tempo di compilazione

C. 1Exception in thread main MyExc3

D. 1... (ciclo infinito)

E. Nessuna delle precedenti

10. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

A. s = (Error []) r;

B. r = (Object [] []) b;

C. r = b;

D. r = (Object [] []) s;

E. r = s;

11. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2){  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```

if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}

```

- A. 001  
B. 101  
C. 010  
D. 000  
E. 111
- 

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
B. public int f( int j)  
C. private void f( int j)  
D. public static void f( int j)  
E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java  
1. package pkgA;  
2. public class A {  
3. private int i = 1;  
4. int getI() { return i; }  
5. }

B.java  
1. package pkgB;  
2. import pkgA.A;  
3. public class B extends A {  
4. public static void main(String[] args) {  
5. int a = new A().getI();  
6. int b = new B().getI();  
7. }  
8. }

- A. Nessun errore a tempo di compilazione e di esecuzione  
B. Errore a tempo di compilazione  
C. Errore a tempo di esecuzione alla linea 5 in B.java  
D. Errore a tempo di esecuzione alla linea 6 in B.java
- 

E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 1524Exception in thread main MyExc1  
C. 15  
D. 1524333333... (ciclo infinito)  
E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {

```

```
super("C");
System.out.print("D");
}
}
class Palm extends Tree {
public static void main(String argv[]) {
    System.out.print("A");
    Palm p = new Palm();
}
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 2**

1. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
}
```

```
class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}
```

```
class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

2. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;

- B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

4. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

5. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
```

```

Tree(String s) {
    super("C");
    System.out.print(s);
}
}

class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 1524333333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {

```

```

        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

10. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {

```

```

        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

11. Se nella classe C è dichiarato il metodo:

```
public void f( int i )
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i )
  - B. public int f( int j )
  - C. private void f( int j )
  - D. public static void f( int j )
  - E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```

A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

```

B.java
1. package pkgB;

```

```

2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

### 13. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

### 14. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() /* do something */ {
        public static void main (String[] args) {
            new B().f();
        }
    }
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

### 15. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {

```

```
    throw( new MyExc1() );
}
catch( MyExc3 i ) {
}
catch( MyExc2 v ) {
}
finally {
    System.out.print(4);
}
}
```

- A. 341
- B. 3... (ciclo infinito)
- C. 31
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

16. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 3**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
```

```
void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

3. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}
```

```

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```

A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

```

B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
- 

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv
                           throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

9. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() /* do something */
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1
- B. o2
- C. o3
- D. o1 e o2
- E. Nessuna delle precedenti.

---

10. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente **static** e **final**
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
    }
}

```

```

        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
 B. public int f( int j)  
 C. private void f( int j)  
 D. public static void f( int j)  
 E. Nessuna delle precedenti
- 

13. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {

```

```

        String s = "A";
        void g() {}
    }

    class B extends A {
        String s = "B";
        public void m() {
            System.out.print(
                super.s + s);
        }
        public void g() {
            ((A)this).g();
        }
        public static void main(String[] args) {
            new B().m();
        }
    }
}

```

- A. AB  
 B. BB  
 C. Errore a tempo di compilazione  
 D. ABException in thread main ...  
 E. Nessuna delle precedenti
- 

14. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
 B. ACD  
 C. ABD  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

16. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 4**

1. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
}
```

```
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

3. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

4. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
}
```

```

A a2 = new A("o2");
A a3 = new A("o3");
void f() {
    a1.setRef(a2);
    a2.setRef(a3);
    a3.setRef(a1);
    a1 = a3; a2 = a3;
    m();
}
void m() { /* do something */ }
public static void main (String[] args) {
    new B().f();
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

5. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

7. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

8. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

9. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A
- B. La classe B
- C. La classe C

- D. Tutte le classi
  - E. Nessuna delle classi
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
}
```

```
void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

```
B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
    finally {
        System.out.print(4);
    }
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione

C. 1Exception in thread main MyExc3

- D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
    finally {
        System.out.print(4);
        throw( new MyExc1() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 1524333333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

16. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
```

```
s5 = new String("ab");
q(s3, b3, b4);
}
void q(String s2, Boolean b1, Boolean b2) {
    String s1;
    s1 = new String("abcde");
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
```

---

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 5**

1. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

2. Quali dei tre oggetti `a1`, `a2` o `a3` sono eleggibili per la garbage collection quando il metodo `m` comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}
```

A. o1

- B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

3. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
```

```

    }
    catch( MyExc2 g ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

### 5. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

### 6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
- B. 3... (ciclo infinito)
- C. 31
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

9. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
    }
}
```

```
super.s = s;
}
public static void main(String[] args) {
    new B().m();
}
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

11. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)

E. Nessuna delle precedenti

---

13. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2) {  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

---

14. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        super("C");  
    }
```

```
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( Exception h ) {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 j ) {  
        }  
        catch( MyExc2 x ) {  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(5);  
            throw( new MyExc2() );  
        }  
        catch( Exception s ) {  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

16. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
```

```
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 6**

1. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

1. package pkgA;
2. public class A {
3. private int i = 1;
4. int getI() { return i; }
5. }

B.java

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4. public static void main(String[] args) {
- 5. int a = new A().getI();
- 6. int b = new B().getI(); }

```
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente **static** e **final**
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

6. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
        a3.setRef(a1);  
        a1 = a3; a2 = a3;  
        m();  
    }  
    void m() { /* do something */ }  
    public static void main (String[] args){  
        new B().f();  
    }  
}  
  
A. o1  
B. o2  
C. o3  
D. o1 e o2  
E. Nessuna delle precedenti.
```

---

7. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

8. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

9. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

10. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
}

```

```

    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
        static void m() throws Exception {
            try {
                throw( new MyExc3() );
            }
            catch( MyExc3 c ) {
                throw( new MyExc3() );
            }
            finally {
                System.out.print(4);
            }
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
-

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

---

14. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

15. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
-

16. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
}  
  
}  
  
public void g() {  
    super.g();  
    super.s = s;  
}  
public static void main(String[] args) {  
    new B().m();  
}
```

---

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 7**

1. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
```

```
void f() {
    a1.setRef(a2);
    a2.setRef(a3);
    a3.setRef(a1);
    a1 = a3; a2 = a3;
    m();
}
void m() { /* do something */ }
public static void main (String[] args) {
    new B().f();
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

3. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

4. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente static e final
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

5. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione  
B. Errore a tempo di compilazione  
C. Errore a tempo di esecuzione alla linea 5 in B.java  
D. Errore a tempo di esecuzione alla linea 6 in B.java  
E. Nessuna delle precedenti
- 

6. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
}
```

```
class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}
```

```
class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
```

```
        return s;
    }
}
```

- A. La classe A  
B. La classe B  
C. La classe C  
D. Tutte le classi  
E. Nessuna delle classi
- 

7. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;  
B. d = (String) c;  
C. d = (String) u;  
D. u = (Integer) d;  
E. Nessuno dei precedenti
- 

8. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)  
B. public String f(byte k)  
C. protected String f(int k)  
D. String f(int i) throws Exception  
E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}
```

```
abstract class A implements I {
    String s = "A";
    void g() {}
}
```

```
public class B extends A {
    String s = "B";
    public void m() {
```

```

        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

10. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
- B. 101
- C. 010

D. 000

E. 111

---

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {

```

```

public static void main(String argv[]) {
    System.out.print("A");
    Palm p = new Palm();
}
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;

- B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {

```

```
        System.out.print(1);
    }
    catch( MyExc2 e ) {
        throw( new Error() );
    }
    catch( MyExc3 f ) {
        System.out.print(2);
    }
}
static void n() {
    try {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc1 b ) {
        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 8

1. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente `static` e `final`
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

2. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

B.java

```
1. package pkgB;  
2. import pkgA.A;  
3. public class B extends A {  
4.     public static void main(String[] args) {  
5.         int a = new A().getI();  
6.         int b = new B().getI();  
7.     }  
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}
```

```
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}
```

```
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
}
```

```

public void g() {
    ((A)this).g();
}
public static void main(String[] args) {
    new B().m();
}
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {

```

```

        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

8. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {

```

```

private String s;
public void sets(String s) {
    this.s = s;
}
public String gets() {
    return s;
}
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

9. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

12. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 1524333333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {

```

```

        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

```

```

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
    }
}

```

```
a1 = a3; a2 = a3;  
m();  
}  
void m() { /* do something */ }  
public static void main (String[] args){  
    new B().f();  
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

16. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 9**

1. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
```

```
void f() {
    a1.setRef(a2);
    a2.setRef(a3);
    a3.setRef(a1);
    a1 = a3; a2 = a3;
    m();
}
void m() { /* do something */ }
public static void main (String[] args) {
    new B().f();
}
```

- A. o1
- B. o2
- C. o3
- D. o1 e o2
- E. Nessuna delle precedenti.

---

3. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
}
public static void main(String[] args) {
    new B().m();
}
```

- A. AB

- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

---

4. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;
- D. u = (Integer) d;
- E. Nessuno dei precedenti

---

5. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2){
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

---

6. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;
- D. r = (Object [] []) s;
- E. r = s;

---

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant {  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}
```

```
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void setS(String s) {  
        this.s = s;  
    }  
    private String getS() {  
        return s;  
    }  
}  
  
class Z {  
    private String s;  
    public void setS(String s) {  
        this.s = s;  
    }  
    public String getS() {  
        return s;  
    }  
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends Error { }  
class MyExc3 extends Error { }  
public class C1 {  
    public static void main(String[] argv) {  
        try {  
            n();  
        }  
        catch( MyExc1 z ) {  
            System.out.print(1);  
        }  
    }  
}
```

```

    }
    catch( MyExc2 e ) {
        throw( new Error() );
    }
    catch( MyExc3 f ) {
        System.out.print(2);
    }
}
static void n() {
    try {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc1 b ) {
        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 341  
B. 3... (ciclo infinito)  
C. 31  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
B. public int f( int j)  
C. private void f( int j)  
D. public static void f( int j)  
E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*  
B. Un attributo può essere contemporaneamente static e final  
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- 

- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading  
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

14. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)  
B. public String f(byte k)  
C. protected String f(int k)  
D. String f(int i) throws Exception  
E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

- B. Errore a tempo di compilazione  
C. Errore a tempo di esecuzione alla linea 5 in B.java  
D. Errore a tempo di esecuzione alla linea 6 in B.java  
E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );

```

- ```

        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

```
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 10

1. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

2. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc2 g ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 c ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java  
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

```

B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione  
 B. Errore a tempo di compilazione  
 C. Errore a tempo di esecuzione alla linea 5 in B.java  
 D. Errore a tempo di esecuzione alla linea 6 in B.java  
 E. Nessuna delle precedenti
- 

5. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}

```

- A. La classe A  
 B. La classe B  
 C. La classe C  
 D. Tutte le classi  
 E. Nessuna delle classi
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
}

```

```

    }
    catch( MyExc3 j ) {
    }
    catch( MyExc2 x ) {
    }
    finally {
        System.out.print(4);
        throw( new MyExc1() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}

```

- A. Errore a tempo di compilazione  
B. 1524Exception in thread main MyExc1  
C. 15  
D. 152433333... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 8. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;  
B. r = (Object [] []) b;  
C. r = b;  
D. r = (Object [] []) s;  
E. r = s;
- 

#### 9. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

```

```

class B {

```

```

    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args){
        new B().f();
    }
}

```

- A. o1  
B. o2  
C. o3  
D. o1 e o2  
E. Nessuna delle precedenti.
- 

#### 10. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
B. BB  
C. Errore a tempo di compilazione  
D. ABException in thread main ...  
E. Nessuna delle precedenti
- 

#### 11. Se nella classe C è dichiarato il metodo:

---

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente **static** e **final**
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

14. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2){  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

---

15. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {
```

```
public static void main(String argv[]) {  
    System.out.print("A");  
    Palm p = new Palm();  
}  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

Object c;

---

```
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 11**

1. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
```

```
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 341
- B. 3... (ciclo infinito)
- C. 31
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}
```

```

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

4. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1
- B. o2

- C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

5. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente **static** e **final**
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

7. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

8. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

#### 9. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

#### 10. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

#### 11. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

1. package pkgA;
2. public class A {
3. private int i = 1;
4. int getI() { return i; }
5. }

B.java

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4. public static void main(String[] args) {
5. int a = new A().getI();
6. int b = new B().getI();
7. }
8. }

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

A. Errore a tempo di compilazione

B. 1524Exception in thread main MyExc1

C. 15

D. 1524333333... (ciclo infinito)

E. Nessuna delle precedenti

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}
```

```
public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

14. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

A. private void f( float i)

B. public int f( int j)

C. private void f( int j)

D. public static void f( int j)

E. Nessuna delle precedenti

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
    }
}
```

```
    }
    finally {
        System.out.print(4);
    }
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;
- D. r = (Object [] []) s;
- E. r = s;

16. Date le dichiarazioni:

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 12**

1. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3

- D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 1524333333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

6. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)

- B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente static e final
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

8. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

9. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
}

```

- A. AB
- B. BB
- C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

#### 11. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

#### 12. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;
- D. u = (Integer) d;
- E. Nessuno dei precedenti

---

13. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2){  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends Error { }  
class MyExc3 extends Error { }  
public class C1 {  
    public static void main(String[] argv){  
        try {  
            n();  
        }  
        catch( MyExc1 z ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 e ) {  
            throw( new Error() );  
        }  
    }  
}
```

```
}  
catch( MyExc3 f ) {  
    System.out.print(2);  
}  
}  
static void n() {  
    try {  
        System.out.print(3);  
        throw( new MyExc1() );  
    }  
    catch( MyExc1 b ) {  
        throw( new MyExc1() );  
    }  
    catch( MyExc3 i ) {  
    }  
    catch( MyExc2 v ) {  
    }  
    finally {  
        System.out.print(4);  
    }  
}  
}
```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

B.java

```
1. package pkgB;  
2. import pkgA.A;  
3. public class B extends A {  
4.     public static void main(String[] args){  
5.         int a = new A().getI();  
6.         int b = new B().getI();  
7.     }  
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
-

16. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}
```

```
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 13**

1. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
}
```

```
static void m() throws Exception {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
    }
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

3. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

4. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {

```

```

        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

7. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```

A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

```

B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione  
 B. Errore a tempo di compilazione  
 C. Errore a tempo di esecuzione alla linea 5 in B.java  
 D. Errore a tempo di esecuzione alla linea 6 in B.java  
 E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;  
 B. d = (String) c;  
 C. d = (String) u;  
 D. u = (Integer) d;  
 E. Nessuno dei precedenti
- 

9. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

```

```

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
 B. BB  
 C. Errore a tempo di compilazione  
 D. ABException in thread main ...  
 E. Nessuna delle precedenti
- 

10. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
 B. public int f( int j)  
 C. private void f( int j)  
 D. public static void f( int j)  
 E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

```

        throw( new MyExc1() );
    }
}

static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 1524333333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

12. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

13. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente static e final
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

14. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
        a3.setRef(a1);  
    }  
    a1 = a3; a2 = a3;  
    m();  
}  
void m() { /* do something */ }  
public static void main (String[] args) {  
    new B().f();  
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 14**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
```

```
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

3. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

---

5. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

---

6. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente static e final
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

7. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}
```

```
}
```

```
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

8. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

9. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java  
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

```
B.java
```

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione  
 B. Errore a tempo di compilazione  
 C. Errore a tempo di esecuzione alla linea 5 in B.java  
 D. Errore a tempo di esecuzione alla linea 6 in B.java  
 E. Nessuna delle precedenti
- 

10. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1  
 B. o2  
 C. o3  
 D. o1 e o2  
 E. Nessuna delle precedenti.
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3  
 B. Errore a tempo di compilazione  
 C. 1Exception in thread main MyExc3  
 D. 1... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
}

```

```

        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

### 13. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
 B. BB  
 C. Errore a tempo di compilazione  
 D. ABException in thread main ...  
 E. Nessuna delle precedenti
- 

### 14. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
 B. BB  
 C. Errore a tempo di compilazione  
 D. ABException in thread main ...  
 E. Nessuna delle precedenti
- 

### 15. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;  
 B. d = (String) c;  
 C. d = (String) u;  
 D. u = (Integer) d;  
 E. Nessuno dei precedenti
- 

### 16. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
}

```

```
private String s4 = new String("ab");
private String s5;
void p() {
    String s3;
    s3 = new String("abcde");
    b4 = b3;
    s5 = new String("ab");
    q(s3, b3, b4);
}
void q(String s2, Boolean b1, Boolean b2) {
    String s1;
    s1 = new String("abcde");
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- 
- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 15**

1. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
```

```
String s = "B";
public void m() {
    System.out.print(
        super.s + s);
}
public void g() {
    super.g();
    super.s = s;
}
public static void main(String[] args) {
    new B().m();
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

3. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception

E. Nessuna delle precedenti.

---

5. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2) {  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

---

6. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java  
1. package pkgA;  
2. public class A {  
3. private int i = 1;  
4. int getI() { return i; }  
5. }

B.java  
1. package pkgB;  
2. import pkgA.A;

```
3. public class B extends A {  
4.     public static void main(String[] args) {  
5.         int a = new A().getI();  
6.         int b = new B().getI();  
7.     }  
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

---

7. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente **static** e **final**
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

8. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant {  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( Exception h ) {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 j ) {  
        }  
        catch( MyExc2 x ) {  
        }  
    }  
}
```

```
}  
finally {  
    System.out.print(4);  
    throw( new MyExc1() );  
}  
}  
static void q() throws Exception {  
    try {  
        System.out.print(5);  
        throw( new MyExc2() );  
    }  
    catch( Exception s ) {  
    }  
}  
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends Error {}  
class MyExc3 extends Error {}  
public class C1 {  
    public static void main(String[] argv) {  
        try {  
            n();  
        }  
        catch( MyExc1 z ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 e ) {  
            throw( new Error() );  
        }  
        catch( MyExc3 f ) {  
            System.out.print(2);  
        }  
    }  
}
```

```

    }
}

static void n() {
    try {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc1 b ) {
        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 341  
B. 3... (ciclo infinito)  
C. 31  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

### 13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3  
B. Errore a tempo di compilazione
- 

C. 1Exception in thread main MyExc3

- D. 1... (ciclo infinito)  
E. Nessuna delle precedenti
- 

### 14. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;  
B. r = (Object [] []) b;  
C. r = b;  
D. r = (Object [] []) s;  
E. r = s;
- 

### 15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1  
B. o2  
C. o3  
D. o1 e o2  
E. Nessuna delle precedenti.
-

16. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}
```

```
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 16**

1. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

2. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

3. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]){  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( Exception h ) {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
    }  
}
```

```

        catch( MyExc3 j ) {
    }
    catch( MyExc2 x ) {
    }
    finally {
        System.out.print(4);
        throw( new MyExc1() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

1. package pkgA;
2. public class A {
3. private int i = 1;
4. int getI() { return i; }
5. }

B.java

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4. public static void main(String[] args) {
5. int a = new A().getI();
6. int b = new B().getI();
7. }
8. }

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
    }
}

```

```

    }
    catch( MyExc2 g ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 14Exception in thread main MyExc3  
B. Errore a tempo di compilazione  
C. 1Exception in thread main MyExc3  
D. 1... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
B. BB  
C. Errore a tempo di compilazione  
D. ABException in thread main ...  
E. Nessuna delle precedenti
- 

#### 10. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;  
B. r = (Object [] []) b;  
C. r = b;  
D. r = (Object [] []) s;  
E. r = s;
- 

#### 11. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A  
B. La classe B  
C. La classe C  
D. Tutte le classi  
E. Nessuna delle classi
- 

#### 12. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Se nella classe C è dichiarato il metodo:

```
public void f( int i )
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i )
  - B. public int f( int j )
  - C. private void f( int j )
  - D. public static void f( int j )
  - E. Nessuna delle precedenti
- 

14. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

15. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente **static** e **final**
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
-

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
    }  
}
```

```
a2.setRef(a3);  
a3.setRef(a1);  
a1 = a3; a2 = a3;  
m();  
}  
void m() { /* do something */ }  
public static void main (String[] args){  
    new B().f();  
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

Prova n. 17

1. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
```

```

try {
    System.out.print(1);
    m();
    System.out.print(2);
}
catch( MyExc2 g ) {
    System.out.print(3);
}
}

static void m() throws Exception {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 14Exception in thread main MyExc3  
B. Errore a tempo di compilazione  
C. 1Exception in thread main MyExc3  
D. 1... (ciclo infinito)  
E. Nessuna delle precedenti
- 

5. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}

```

- A. La classe A  
B. La classe B

- C. La classe C  
D. Tutte le classi  
E. Nessuna delle classi
- 

6. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
B. ACD  
C. ABD  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

7. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
B. public int f( int j)  
C. private void f( int j)  
D. public static void f( int j)  
E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
    }
}

```

```

        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A

- B. ACD  
 C. ABD  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 10. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1  
 B. o2  
 C. o3  
 D. o1 e o2  
 E. Nessuna delle precedenti.
- 

#### 11. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*  
 B. Un attributo può essere contemporaneamente `static` e `final`  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Quando due metodi hanno lo stesso nome non sempre si ha overloading  
 E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

12. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

14. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

15. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
}
```

```
public void g() {  
    super.g();  
    super.s = s;  
}  
public static void main(String[] args) {  
    new B().m();  
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 18**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
```

```
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 341
- B. 3... (ciclo infinito)
- C. 31
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
}
```

```

Plant(String s) {
    System.out.print(s);
}

class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}

class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }

public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }

    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)

E. Nessuna delle precedenti

---

#### 5. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}

class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}

class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 6. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

#### 7. Se nella classe C è dichiarato il metodo:

```

String f(int i)

```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)

- D. `String f(int i) throws Exception`  
E. Nessuna delle precedenti.
- 

8. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

9. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `u = (Integer) c;`
  - B. `d = (String) c;`
  - C. `d = (String) u;`
  - D. `u = (Integer) d;`
  - E. Nessuno dei precedenti
- 

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
}
```

```
    g();
}
public void g() {
    super.g();
    super.s = s;
}
public static void main(String[] args) {
    new B().m();
}
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
}
public static void main(String[] args) {
    new B().m();
}
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

12. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

```

```

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

13. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {

```

```

            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
    A. 001
    B. 101
    C. 010
    D. 000
    E. 111

```

---

14. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

1. package pkgA;
2. public class A {
3. private int i = 1;
4. int getI() { return i; }
5. }

B.java

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4. public static void main(String[] args) {
5. int a = new A().getI();
6. int b = new B().getI();
7. }
8. }

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
}

```

```
A a2 = new A("o2");
A a3 = new A("o3");
void f() {
    a1.setRef(a2);
    a2.setRef(a3);
    a3.setRef(a1);
    a1 = a3; a2 = a3;
    m();
}
void m() { /* do something */ }
public static void main (String[] args) {
    new B().f();
}
```

- A. o1
- B. o2
- C. o3
- D. o1 e o2

E. Nessuna delle precedenti.

---

16. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 19

1. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

2. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant {  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant {  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}  
  
A. A  
B. ACD  
C. ABD  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
```

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends Error { }  
class MyExc3 extends Error { }  
public class C1 {  
    public static void main(String[] argv) {  
        try {  
            n();  
        }  
        catch( MyExc1 z ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 e ) {  
            throw( new Error() );  
        }  
        catch( MyExc3 f ) {  
            System.out.print(2);  
        }  
    }  
}
```

```

}
static void n() {
    try {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc1 b ) {
        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- ```

}
A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 1524333333... (ciclo infinito)
E. Nessuna delle precedenti

```
- 

#### 6. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

#### 7. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

```

```

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```

1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

B.java

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

10. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

```

```

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente static e final
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

14. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}
```

```
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");
```

```
A a3 = new A("o3");  
void f() {  
    a1.setRef(a2);  
    a2.setRef(a3);  
    a3.setRef(a1);  
    a1 = a3; a2 = a3;  
    m();  
}  
void m() { /* do something */ }  
public static void main (String[] args){  
    new B().f();  
}
```

---

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

15. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void sets(String s) {  
        this.s = s;  
    }  
    private String gets() {  
        return s;  
    }  
}  
  
class Z {  
    private String s;  
    public void sets(String s) {  
        this.s = s;  
    }  
    public String gets() {  
        return s;  
    }  
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );

```

- ```

        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

```
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 20**

1. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
}
```

```
g();
}
public void g() {
    ((A)this).g();
}
public static void main(String[] args) {
    new B().m();
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args){
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

#### 5. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

#### 6. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

#### 7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
    }
}

```

```

    }
    catch( MyExc2 e ) {
        throw( new Error() );
    }
    catch( MyExc3 f ) {
        System.out.print(2);
    }
}
static void n() {
    try {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc1 b ) {
        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 9. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
A. 001
B. 101
C. 010
D. 000
E. 111

```

---

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 11. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( Exception h ) {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 j ) {  
        }  
        catch( MyExc2 x ) {  
        }  
    }  
}
```

```
        }  
    }  
    finally {  
        System.out.print(4);  
        throw( new MyExc1() );  
    }  
}  
static void q() throws Exception {  
    try {  
        System.out.print(5);  
        throw( new MyExc2() );  
    }  
    catch( Exception s ) {  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

14. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i )
  - B. public int f( int j )
  - C. private void f( int j )
  - D. public static void f( int j )
  - E. Nessuna delle precedenti
- 

15. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente **static** e **final**
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

16. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void setS(String s) {  
        this.s = s;  
    }  
    private String getS() {  
        return s;  
    }  
}  
  
class Z {  
    private String s;
```

```
    public void setS(String s) {  
        this.s = s;  
    }  
    public String getS() {  
        return s;  
    }  
}
```

---

- A. La classe A
- B. La classe B
- C. La classe C
- D. Tutte le classi
- E. Nessuna delle classi



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 21**

1. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args){
        new B().f();
    }
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

2. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)

- D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

3. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}
```

```
abstract class A implements I {
    String s = "A";
    void g() {}
```

```

}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

5. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

6. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. `private void f( float i)`
  - B. `public int f( int j)`
  - C. `private void f( int j)`
  - D. `public static void f( int j)`
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {

```

```

        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;  
 B. r = (Object [] []) b;  
 C. r = b;  
 D. r = (Object [] []) s;  
 E. r = s;
- 

10. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
 B. ACD  
 C. ABD  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

11. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
 B. ACD  
 C. ABD  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
    finally {
    }
}

```

```

        System.out.print(4);
        throw( new MyExc1() );
    }

static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}
}

```

- A. Errore a tempo di compilazione  
B. 1524Exception in thread main MyExc1  
C. 15  
D. 152433333... (ciclo infinito)  
E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;  
B. d = (String) c;  
C. d = (String) u;  
D. u = (Integer) d;  
E. Nessuno dei precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```

A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione  
B. Errore a tempo di compilazione  
C. Errore a tempo di esecuzione alla linea 5 in B.java  
D. Errore a tempo di esecuzione alla linea 6 in B.java  
E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001  
B. 101  
C. 010  
D. 000  
E. 111
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );

```

- ```

        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

```
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 22

1. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente `static` e `final`
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i )
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. `private void f( float i )`
- B. `public int f( int j )`
- C. `private void f( int j )`
- D. `public static void f( int j )`
- E. Nessuna delle precedenti

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente `f` (overriding).

- A. `public void f(int j)`
- B. `public String f(byte k)`
- C. `protected String f(int k)`
- D. `String f(int i) throws Exception`
- E. Nessuna delle precedenti.

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

5. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

#### 6. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
    }
}

```

```

    Palm p = new Palm();
}
A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

#### 7. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```

1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

B.java

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args){
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
}

```

```

    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {

```

```

    public static void main(String [] argv)
                            throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

- 
12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
        a3.setRef(a1);  
        a1 = a3; a2 = a3;  
        m();  
    }  
    void m() { /* do something */ }  
    public static void main (String[] args){  
        new B().f();  
    }  
}
```

- A. o1
- B. o2
- C. o3
- D. o1 e o2
- E. Nessuna delle precedenti.

- 
13. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;
- D. r = (Object [] []) s;
- E. r = s;

- 
14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends Error { }  
class MyExc3 extends Error { }  
public class C1 {  
    public static void main(String[] argv) {  
        try {  
            n();  
        }  
        catch( MyExc1 z ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 e ) {  
            throw( new Error() );  
        }  
        catch( MyExc3 f ) {  
            System.out.print(2);  
        }  
    }  
    static void n() {  
        try {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc1 b ) {  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 i ) {  
        }  
        catch( MyExc2 v ) {  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 341
- B. 3... (ciclo infinito)
- C. 31
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

- 
15. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;
- D. u = (Integer) d;
- E. Nessuno dei precedenti

---

16. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void sets(String s) {  
        this.s = s;  
    }  
    private String gets() {  
        return s;  
    }  
}
```

```
class Z {  
    private String s;  
    public void sets(String s) {  
        this.s = s;  
    }  
    public String gets() {  
        return s;  
    }  
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 23**

1. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
}
```

```
g();
}
public void g() {
    super.g();
    super.s = s;
}
public static void main(String[] args) {
    new B().m();
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 14Exception in thread main MyExc3

- B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( Exception h ) {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 j ) {  
        }  
        catch( MyExc2 x ) {  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(5);  
            throw( new MyExc2() );  
        }  
        catch( Exception s ) {  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 1524333333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

8. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

9. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

```

```

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
    }
}

```

```

s1 = new String("abcde");
if(s2 == s1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

**12. Date le dichiarazioni:**

```

Object [] b;
Object [][] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

**13. Se nella classe C è dichiarato il metodo:**

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

**14. Qual è il risultato della compilazione ed esecuzione dei seguenti file?**

A.java

```

1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

B.java

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

**15. Qual è l'output di questo codice?**

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {

```

```
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}
```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente **static** e **final**
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 24**

1. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `u = (Integer) c;`
  - B. `d = (String) c;`
  - C. `d = (String) u;`
  - D. `u = (Integer) d;`
  - E. Nessuno dei precedenti
- 

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
}
public static void main (String[] args) {
    new B().f();
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

3. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
}
```

```

    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 6. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

#### 7. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

#### 8. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

```
B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

---

#### 9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
```

```
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 10. Se nella classe C è dichiarato il metodo:

```
public void f( int i )
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i )
  - B. public int f( int j )
  - C. private void f( int j )
  - D. public static void f( int j )
  - E. Nessuna delle precedenti
- 

#### 11. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 12. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

### 13. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione

### E. Nessuna delle precedenti

---

### 14. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

### 15. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

```

```

abstract class A implements I {
    String s = "A";
    void g() {}
}

```

```
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...

E. Nessuna delle precedenti

---

16. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 25**

1. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
}
public static void main (String[] args) {
    new B().f();
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
```

```
class Tree extends Plant {
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Se nella classe C è dichiarato il metodo:

String f(int i)

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

4. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente static e final
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

6. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}
```

```
class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {

```

```

        n();
    }
    catch( MyExc1 z ) {
        System.out.print(1);
    }
    catch( MyExc2 e ) {
        throw( new Error() );
    }
    catch( MyExc3 f ) {
        System.out.print(2);
    }
}
static void n() {
    try {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc1 b ) {
        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
}

```

```

    }
    public static void main(String[] args) {
        new B().m();
    }
}

A. AB
B. BB
C. Errore a tempo di compilazione
D. ABException in thread main ...
E. Nessuna delle precedenti

```

---

#### 10. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
 B. ACD  
 C. ABD  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 11. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
    }
}

```

```

s1 = new String("abcde");
if(s2 == s1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

## 12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- }
  - }
  - A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

## 13. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

## 14. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

## 15. Se nella classe C è dichiarato il metodo:

```

public void f( int i)

```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

---

16. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
```

```
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 26**

1. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)

- E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 1524333333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

4. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A  
B. La classe B  
C. La classe C  
D. Tutte le classi  
E. Nessuna delle classi
- 

#### 5. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;  
B. r = (Object [] []) b;  
C. r = b;  
D. r = (Object [] []) s;  
E. r = s;
- 

#### 6. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

```

```

    }

    class B extends A {
        String s = "B";
        public void m() {
            System.out.print(
                super.s + s);
        }
        public void g() {
            ((A)this).g();
        }
        public static void main(String[] args) {
            new B().m();
        }
    }
}

```

- A. AB  
B. BB  
C. Errore a tempo di compilazione  
D. ABException in thread main ...  
E. Nessuna delle precedenti
- 

#### 7. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
B. ACD  
C. ABD  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

## 9. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A

- B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

## 10. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

- A.java
1. package pkgA;
  2. public class A {
  3. private int i = 1;
  4. int getI() { return i; }
  5. }
- B.java
1. package pkgB;
  2. import pkgA.A;
  3. public class B extends A {
  4. public static void main(String[] args){
  5. int a = new A().getI();
  6. int b = new B().getI();
  7. }
  8. }
- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

## 11. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2){
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {

```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

12. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

13. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*

- B. Un attributo può essere contemporaneamente static e final
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
}
```

- A. 341  
B. 3... (ciclo infinito)  
C. 31  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 27**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args){
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti
```

---

3. Se nella classe C è dichiarato il metodo:

public void f( int i)

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

4. Se nella classe C è dichiarato il metodo:

String f(int i)

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)

- B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

5. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

6. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        super.g();  
        super.s = s;  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant {  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2) {  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        }  
    }  
}
```

```

    } else {
        System.out.print(0);
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

9. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;

- C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

11. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

12. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

13. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

#### 14. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

#### 15. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );

```

- ```

        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

```
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 28**

1. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
}
```

```
class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}
```

```
class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

2. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;

- B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

3. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

4. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}
```

```
class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
}
```

```
void m() { /* do something */ }
public static void main (String[] args) {
    new B().f();
}
```

- A. o1
- B. o2
- C. o3
- D. o1 e o2
- E. Nessuna delle precedenti.

---

5. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente **static** e **final**
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

6. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

---

7. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

---

8. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```

---

9. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti
```

---

10. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
```

```
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 341
- B. 3... (ciclo infinito)
- C. 31

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
```

```
public static void main(String argv[]) {
    System.out.print("A");
    Palm p = new Palm();
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
    }
}
```

```
}

catch( Exception h ) {
    System.out.print(3);
    throw( new MyExc1() );
}
catch( MyExc3 j ) {
}
catch( MyExc2 x ) {
}
finally {
    System.out.print(4);
    throw( new MyExc1() );
}

static void q() throws Exception {
    try {

}
System.out.print(5);
throw( new MyExc2() );
}
catch( Exception s ) {
}
}

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 152433333... (ciclo infinito)
E. Nessuna delle precedenti
```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 29

1. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

3. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
- B. 101
- C. 010
- D. 000

E. 111

---

4. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

5. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
}
```

```

void m() { /* do something */ }
public static void main (String[] args) {
    new B().f();
}
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

8. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

9. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A

- B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

10. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. `private void f( float i)`
  - B. `public int f( int j)`
  - C. `private void f( int j)`
  - D. `public static void f( int j)`
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**14. Qual è l'output di questo programma?**

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**15. Date le dichiarazioni:**

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;

- D. `u = (Integer) d;`
  - E. Nessuno dei precedenti
- 

16. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

B.java

```
1. package pkgB;  
2. import pkgA.A;
```

```
3. public class B extends A {  
4.     public static void main(String[] args){  
5.         int a = new A().getI();  
6.         int b = new B().getI();  
7.     }  
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 30**

1. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
}
```

```
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
```

```

        System.out.print(1);
    }
    catch( MyExc2 e ) {
        throw( new Error() );
    }
    catch( MyExc3 f ) {
        System.out.print(2);
    }
}
static void n() {
    try {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc1 b ) {
        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
 B. BB  
 C. Errore a tempo di compilazione  
 D. ABException in thread main ...  
 E. Nessuna delle precedenti
- 

#### 6. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
 B. public int f( int j)  
 C. private void f( int j)  
 D. public static void f( int j)  
 E. Nessuna delle precedenti
- 

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 1524333333... (ciclo infinito)
- E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;
- D. r = (Object [] []) s;
- E. r = s;

---

9. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente static e final
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

10. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

B.java

```
1. package pkgB;  
2. import pkgA.A;  
3. public class B extends A {  
4.     public static void main(String[] args) {  
5.         int a = new A().getI();  
6.         int b = new B().getI();  
7.     }  
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc2 g ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 c ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

---

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

13. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
- B. ACD

C. ABD

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

15. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

16. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2) {  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

---

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 31

1. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc2 g ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 c ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

4. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

---

5. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2){  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

6. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;

- D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

7. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        super.g();  
        super.s = s;  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
}
```

```

    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```

1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

B.java

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {

```

```

            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }

```

```

        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
    catch( MyExc3 j ) {
    }
    catch( MyExc2 x ) {
    }
    finally {
        System.out.print(4);
        throw( new MyExc1() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}
}

```

- A. Errore a tempo di compilazione  
B. 1524Exception in thread main MyExc1  
C. 15  
D. 1524333333... (ciclo infinito)  
E. Nessuna delle precedenti
- 

13. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
}
public static void main (String[] args) {
    new B().f();
}

```

- A. o1  
B. o2  
C. o3  
D. o1 e o2  
E. Nessuna delle precedenti.
- 

14. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}

```

- A. La classe A  
B. La classe B  
C. La classe C  
D. Tutte le classi  
E. Nessuna delle classi
- 

15. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {

```

```
public static void main(String argv[]) {  
    System.out.print("A");  
    Palm p = new Palm();  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

16. Date le dichiarazioni:

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 32**

1. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
}
public static void main (String[] args) {
    new B().f();
}
```

- A. o1
- B. o2
- C. o3

- D. o1 e o2

- E. Nessuna delle precedenti.
- 

3. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

4. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)

- D. public static void f( int j)  
E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;  
B. d = (String) c;  
C. d = (String) u;  
D. u = (Integer) d;  
E. Nessuno dei precedenti
- 

6. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB  
B. BB  
C. Errore a tempo di compilazione  
D. ABException in thread main ...  
E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( Exception h ) {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 j ) {  
        }  
        catch( MyExc2 x ) {  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(5);  
            throw( new MyExc2() );  
        }  
        catch( Exception s ) {  
        }  
    }  
}
```

- A. Errore a tempo di compilazione  
B. 1524Exception in thread main MyExc1  
C. 15  
D. 152433333... (ciclo infinito)  
E. Nessuna delle precedenti
- 

8. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(
```

```

        super.s + s);
    g();
}
public void g() {
    super.g();
    super.s = s;
}
public static void main(String[] args) { new B().m(); }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
- 

C. 31

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```

1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

B.java

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

11. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente **static** e **final**
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

12. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

### 13. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
    }
}

```

```

    Palm p = new Palm();
}
A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

### 14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

### 15. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;

- C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

16. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
    }  
}
```

```
        System.out.print(s);  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]){  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 33**

1. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente `static` e `final`
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
```

```
} }
```

A. 341  
B. 3... (ciclo infinito)  
C. 31  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
    finally {
        System.out.print(4);
        throw( new MyExc1() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
```

```
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

4. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

5. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
```

```
String s = "B";
public void m() {
    System.out.print(
        super.s + s);
}
public void g() {
    super.g();
    super.s = s;
}
public static void main(String[] args) {
    new B().m();
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

```
B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

8. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. o1
- B. o2

- C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        }
    }
}
```

```

    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 001  
 B. 101  
 C. 010  
 D. 000  
 E. 111
- 

11. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)  
 B. public String f(byte k)  
 C. protected String f(int k)  
 D. String f(int i) throws Exception  
 E. Nessuna delle precedenti.
- 

12. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;  
 B. r = (Object [] []) b;  
 C. r = b;  
 D. r = (Object [] []) s;  
 E. r = s;
- 

13. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)

- B. public int f( int j)  
 C. private void f( int j)  
 D. public static void f( int j)  
 E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3  
 B. Errore a tempo di compilazione  
 C. 1Exception in thread main MyExc3  
 D. 1... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
}

```

```
    g();
}
public void g() {
    ((A)this).g();
}
public static void main(String[] args) {
    new B().m();
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

---

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;
- D. u = (Integer) d;
- E. Nessuno dei precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 34**

1. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente **static** e **final**
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. **private void f( float i)**
  - B. **public int f( int j)**
  - C. **private void f( int j)**
  - D. **public static void f( int j)**
  - E. Nessuna delle precedenti
- 

3. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
```

```
s1 = new String("abcde");
if(s2 == s1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

4. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. **s = (Error []) r;**
  - B. **r = (Object [] []) b;**
  - C. **r = b;**
  - D. **r = (Object [] []) s;**
  - E. **r = s;**
- 

5. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args){
        new B().f();
    }
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

8. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
}

```

```
public String getS() {  
    return s;  
}  
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

9. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

B.java

```
1. package pkgB;  
2. import pkgA.A;  
3. public class B extends A {  
4.     public static void main(String[] args) {  
5.         int a = new A().getI();  
6.         int b = new B().getI();  
7.     }  
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc2 g ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 c ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        super.g();  
        super.s = s;  
    }  
}
```

```

}
public static void main(String[] args) {
    new B().m();
}
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

```

        throw( new MyExc1() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}

```

- 
- A. Errore a tempo di compilazione  
B. 1524Exception in thread main MyExc1  
C. 15  
D. 152433333... (ciclo infinito)  
E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 35**

1. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

2. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends Error {}  
class MyExc3 extends Error {}  
public class C1 {  
    public static void main(String[] argv) {  
        try {  
            n();  
        }  
        catch( MyExc1 z ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 e ) {  
            throw( new Error() );  
        }  
    }  
}
```

```
    catch( MyExc3 f ) {  
        System.out.print(2);  
    }  
}  
static void n() {  
    try {  
        System.out.print(3);  
        throw( new MyExc1() );  
    }  
    catch( MyExc1 b ) {  
        throw( new MyExc1() );  
    }  
    catch( MyExc3 i ) {  
    }  
    catch( MyExc2 v ) {  
    }  
    finally {  
        System.out.print(4);  
    }  
}
```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
}
```

```

    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

5. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

6. Quale delle seguenti classi *non* è strettamente encapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

8. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

1. package pkgA;
2. public class A {
3. private int i = 1;
4. int getI() { return i; }
5. }

B.java

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4. public static void main(String[] args) {
5. int a = new A().getI();
6. int b = new B().getI();
7. }
8. }

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java

- D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

10. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

11. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
    }
}

```

```

        System.out.print("D");
    }
}

class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
}

```

```

Plant(String s) {
    System.out.print(s);
}
}

class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}

class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

15. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
-

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
```

```
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 36**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
```

```
private Boolean b4;
private String s4 = new String("ab");
private String s5;
void p() {
    String s3;
    s3 = new String("abcde");
    b4 = b3;
    s5 = new String("ab");
    q(s3, b3, b4);
}
void q(String s2, Boolean b1, Boolean b2) {
    String s1;
    s1 = new String("abcde");
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

3. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)

E. Nessuna delle precedenti

---

4. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
  
class Tree extends Plant {  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente **static** e **final**
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

6. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}
```

```
}  
}  
  
class Tree extends Plant {  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        super.g();  
    }  
}
```

```

        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
 B. BB  
 C. Errore a tempo di compilazione  
 D. ABException in thread main ...  
 E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;  
 B. r = (Object [] []) b;  
 C. r = b;  
 D. r = (Object [] []) s;  
 E. r = s;
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3  
 B. Errore a tempo di compilazione  
 C. 1Exception in thread main MyExc3  
 D. 1... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```

1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

B.java

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione  
 B. Errore a tempo di compilazione  
 C. Errore a tempo di esecuzione alla linea 5 in B.java  
 D. Errore a tempo di esecuzione alla linea 6 in B.java  
 E. Nessuna delle precedenti
- 

12. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;  
 B. d = (String) c;  
 C. d = (String) u;  
 D. u = (Integer) d;  
 E. Nessuno dei precedenti
-

13. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 341
- B. 3... (ciclo infinito)
- C. 31
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
```

```
public void m() {
    System.out.print(
        super.s + s);
    g();
}
public void g() {
    ((A)this).g();
}
public static void main(String[] args) {
    new B().m();
}
```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

15. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}
```

- A. La classe A
- B. La classe B
- C. La classe C
- D. Tutte le classi
- E. Nessuna delle classi

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
        a3.setRef(a1);  
    }  
    a1 = a3; a2 = a3;  
    m();  
}  
void m() { /* do something */ }  
public static void main (String[] args) {  
    new B().f();  
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 37**

1. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

2. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
```

```
String s = "B";
public void m() {
    System.out.print(
        super.s + s);
}
public void g() {
    super.g();
    super.s = s;
}
public static void main(String[] args) {
    new B().m();
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

3. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}
```

```
class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
```

```

public static void main (String[] args){
    new B().f();
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

7. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;
- D. u = (Integer) d;
- E. Nessuno dei precedenti

---

8. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant {  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

B.java

```
1. package pkgB;  
2. import pkgA.A;  
3. public class B extends A {  
4.     public static void main(String[] args) {  
5.         int a = new A().getI();  
6.         int b = new B().getI();  
7.     }  
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
- 

E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc2 g ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 c ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2) {  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {
```

```

        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

12. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

13. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
}

```

```

public void g() {
    ((A)this).g();
}
public static void main(String[] args) {
    new B().m();
}
}

```

A. AB  
B. BB  
C. Errore a tempo di compilazione  
D. ABException in thread main ...  
E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-

15. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

A. Ai metodi static non si applica il *dynamic method dispatch*

B. Un attributo può essere contemporaneamente static e final

C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri

D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

16. Dire quale delle seguenti affermazioni è falsa:

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 38**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
        static void m() throws Exception {
            try {
                throw( new MyExc3() );
            }
            catch( MyExc3 c ) {
                throw( new MyExc3() );
            }
            finally {
                System.out.print(4);
            }
        }
    }
}
```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
```

```
class Tree extends Plant {
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```

---

3. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}
```

```

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {

```

```

        this.s = s;
    }
    public String gets() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 9. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

#### 10. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

#### 11. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() /* do something */
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

13. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

```

```

    }

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente static e final
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

15. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```

A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

```

B.java
1. package pkgB;

```

```
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

---

16. Se nella classe C è dichiarato il metodo:

String f(int i)

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 39**

1. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
```

```
public static void main(String argv[]) {
    System.out.print("A");
    Palm p = new Palm();
}
}
A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

```

    }
}
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

5. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente **static** e **final**
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
    }
}

```

```

        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
static void m() throws Exception {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

7. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

9. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}
```

```
class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

---

11. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        super.g();  
        super.s = s;  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

12. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

B.java

```
1. package pkgB;  
2. import pkgA.A;  
3. public class B extends A {  
4.     public static void main(String[] args){  
5.         int a = new A().getI();  
6.         int b = new B().getI();  
7.     }  
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

14. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

15. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
    }  
}
```

```
a2.setRef(a3);  
a3.setRef(a1);  
a1 = a3; a2 = a3;  
m();  
}  
void m() { /* do something */ }  
public static void main (String[] args){  
    new B().f();  
}
```

- A. o1  
B. o2  
C. o3  
D. o1 e o2  
E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 40**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
        static void m() throws Exception {
            try {
                throw( new MyExc3() );
            }
            catch( MyExc3 c ) {
                throw( new MyExc3() );
            }
            finally {
                System.out.print(4);
            }
        }
    }
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
```

```
b4 = b3;
s5 = new String("ab");
q(s3, b3, b4);
}
void q(String s2, Boolean b1, Boolean b2) {
    String s1;
    s1 = new String("abcde");
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

4. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente static e final
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

5. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

6. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)

- B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

7. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

8. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```

A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

```

B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
```

```
            catch( Exception h ) {
                System.out.print(3);
                throw( new MyExc1() );
            }
            catch( MyExc3 j ) {
            }
            catch( MyExc2 x ) {
            }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 1524333333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
        a3.setRef(a1);  
        a1 = a3; a2 = a3;  
        m();  
    }  
    void m() { /* do something */ }  
    public static void main (String[] args){  
        new B().f();  
    }  
}
```

- A. o1
- B. o2
- C. o3
- D. o1 e o2
- E. Nessuna delle precedenti.

- 
13. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;
- D. u = (Integer) d;
- E. Nessuno dei precedenti

- 
14. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends Error {}  
class MyExc3 extends Error {}  
public class C1 {  
    public static void main(String[] argv) {  
        try {  
            n();  
        }  
        catch( MyExc1 z ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 e ) {  
            throw( new Error() );  
        }  
        catch( MyExc3 f ) {  
            System.out.print(2);  
        }  
    }  
    static void n() {  
        try {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc1 b ) {  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 i ) {  
        }  
        catch( MyExc2 v ) {  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 341
- B. 3... (ciclo infinito)
- C. 31
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

- 
15. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;
- D. r = (Object [] []) s;
- E. r = s;

16. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}
```

```
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 41**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)

- D. public static void f( int j)

- E. Nessuna delle precedenti

3. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. o1
- B. o2
- C. o3
- D. o1 e o2
- E. Nessuna delle precedenti.

4. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv){
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
    }
}

```

```

        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
}
public static void main(String[] args) {
    new B().m();
}

```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

8. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente `static` e `final`
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

10. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

11. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
- B. 101

- C. 010  
D. 000  
E. 111
- 

12. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

13. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void sets(String s) {  
        this.s = s;  
    }  
    private String gets() {  
        return s;  
    }  
}  
  
class Z {  
    private String s;  
    public void sets(String s) {  
        this.s = s;  
    }  
    public String gets() {  
        return s;  
    }  
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

14. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant {  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}  
  
A. A  
B. ACD  
C. ABD  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
```

---

15. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}  
  
A. AB  
B. BB
```

- C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

16. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
```

```
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 42**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
```

```
b4 = b3;
s5 = new String("ab");
q(s3, b3, b4);
}
void q(String s2, Boolean b1, Boolean b2) {
    String s1;
    s1 = new String("abcde");
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

3. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;
- D. r = (Object [] []) s;
- E. r = s;

4. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void sets(String s) {  
        this.s = s;  
    }  
    private String gets() {  
        return s;  
    }  
}  
  
class Z {  
    private String s;  
    public void sets(String s) {  
        this.s = s;  
    }  
    public String gets() {  
        return s;  
    }  
}
```

- A. La classe A
- B. La classe B
- C. La classe C
- D. Tutte le classi
- E. Nessuna delle classi

5. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD

- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
    }  
}
```

```

    a3.setRef(a1);
    a1 = a3; a2 = a3;
    m();
}
void m() { /* do something */ }
public static void main (String[] args) {
    new B().f();
}
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

8. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

9. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

10. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 1524333333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
 B. BB  
 C. Errore a tempo di compilazione  
 D. ABException in thread main ...  
 E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*  
 B. Un attributo può essere contemporaneamente `static` e `final`  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Quando due metodi hanno lo stesso nome non sempre si ha overloading  
 E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {

```

```

            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

1. package pkgA;
2. public class A {
3. private int i = 1;
4. int getI() { return i; }
5. }

B.java

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4. public static void main(String[] args) {
5. int a = new A().getI();
6. int b = new B().getI();
7. }
8. }

A. Nessun errore a tempo di compilazione e di esecuzione  
 B. Errore a tempo di compilazione

- C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        System.out.print("A");  
    }  
}
```

```
    super("C");  
    System.out.print("D");  
}  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 43**

1. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

2. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
1. package pkgA;
```

```
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A

- B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

5. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
}

```

```

void f() {
    a1.setRef(a2);
    a2.setRef(a3);
    a3.setRef(a1);
    a1 = a3; a2 = a3;
    m();
}
void m() { /* do something */ }
public static void main (String[] args) {
    new B().f();
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

7. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

8. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

A. AB  
B. BB  
C. Errore a tempo di compilazione  
D. ABException in thread main ...  
E. Nessuna delle precedenti

---

9. Dire quale delle seguenti affermazioni è falsa:
- A. Ai metodi **static** non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente **static** e **final**
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

10. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

```

```

    }

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

11. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
}

```

```

    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341  
B. 3... (ciclo infinito)  
C. 31  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;  
B. d = (String) c;  
C. d = (String) u;  
D. u = (Integer) d;  
E. Nessuno dei precedenti
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
    }
}

```

```

    }
    catch( MyExc2 g ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 14Exception in thread main MyExc3  
B. Errore a tempo di compilazione  
C. 1Exception in thread main MyExc3  
D. 1... (ciclo infinito)  
E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
B. ACD  
C. ABD  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

```

        throw( new MyExc1() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}

```

- 
- A. Errore a tempo di compilazione  
B. 1524Exception in thread main MyExc1  
C. 15  
D. 152433333... (ciclo infinito)  
E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 44

1. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

3. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;
- D. r = (Object [] []) s;
- E. r = s;

4. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

- A.java
- ```
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```
- B.java
- ```
1. package pkgB;  
2. import pkgA.A;  
3. public class B extends A {  
4.     public static void main(String[] args) {  
5.         int a = new A().getI();  
6.         int b = new B().getI();  
7.     }  
8. }
```
- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
5. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

```

```

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
    }
}

```

```

        catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

#### 8. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente `static` e `final`

- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

9. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

---

10. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
        a3.setRef(a1);  
        a1 = a3; a2 = a3;  
        m();  
    }  
    void m() { /* do something */ }  
}  
public static void main (String[] args){  
    new B().f();  
}
```

- A. o1  
B. o2  
C. o3

- D. o1 e o2  
E. Nessuna delle precedenti.

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc2 g ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 c ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 14Exception in thread main MyExc3  
B. Errore a tempo di compilazione  
C. 1Exception in thread main MyExc3  
D. 1... (ciclo infinito)  
E. Nessuna delle precedenti

---

12. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
}
```

```

public void g() {
    ((A)this).g();
}
public static void main(String[] args) {
    new B().m();
}
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;

- B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

---

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
    }
}

```

```
    System.out.print(2);
}
catch( Exception h ) {
    System.out.print(3);
    throw( new MyExc1() );
}
catch( MyExc3 j ) {
}
catch( MyExc2 x ) {
}
finally {
    System.out.print(4);
    throw( new MyExc1() );
}
}

static void q() throws Exception {
    try {
```

```
    System.out.print(5);
    throw( new MyExc2() );
}
catch( Exception s ) {
}
}
```

- 
- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 45**

1. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2){
        String s1;
        s1 = new String("abcde");
```

```
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

3. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
}
```

```

public String getS() {
    return s;
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

4. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```

1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

B.java

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}

```

```

}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Se nella classe C è dichiarato il metodo:

String f(int i)

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1

- B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
}
```

```
}
static void m() throws Exception {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
    }
}
```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

10. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

11. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
```

```

        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `s = (Error []) r;`
- B. `r = (Object [] []) b;`
- C. `r = b;`
- D. `r = (Object [] []) s;`
- E. `r = s;`

- A. Ai metodi static non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente static e final
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

16. Dire quale delle seguenti affermazioni è falsa:

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 46**

1. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `s = (Error []) r;`
  - B. `r = (Object [] []) b;`
  - C. `r = b;`
  - D. `r = (Object [] []) s;`
  - E. `r = s;`
- 

5. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

8. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}
```

- }
- A. o1  
B. o2  
C. o3  
D. o1 e o2  
E. Nessuna delle precedenti.
- 

#### 9. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB  
B. BB  
C. Errore a tempo di compilazione  
D. ABException in thread main ...  
E. Nessuna delle precedenti
- 

#### 10. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;

- B. d = (String) c;  
C. d = (String) u;  
D. u = (Integer) d;  
E. Nessuno dei precedenti
- 

#### 11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
        static void m() throws Exception {
            try {
                throw( new MyExc3() );
            }
            catch( MyExc3 c ) {
                throw( new MyExc3() );
            }
            finally {
                System.out.print(4);
            }
        }
    }
}
```

- A. 14Exception in thread main MyExc3  
B. Errore a tempo di compilazione  
C. 1Exception in thread main MyExc3  
D. 1... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 12. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)  
B. public String f(byte k)  
C. protected String f(int k)  
D. String f(int i) throws Exception  
E. Nessuna delle precedenti.
-

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
        }
    }
}
```

```
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}
}

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 1524333333... (ciclo infinito)
E. Nessuna delle precedenti
```

15. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

A. 341
B. 3... (ciclo infinito)
C. 31
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```

16. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void setS(String s) {  
        this.s = s;  
    }  
    private String getS() {  
        return s;  
    }  
}  
  
class Z {
```

```
private String s;  
public void setS(String s) {  
    this.s = s;  
}  
public String getS() {  
    return s;  
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 47**

1. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
    }
}
```

```
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

6. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() /* do something */
}
public static void main (String[] args) {

```

- } new B().f();
  - } A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
        static void m() throws Exception {
            try {
                throw( new MyExc3() );
            }
            catch( MyExc3 c ) {
                throw( new MyExc3() );
            }
            finally {
                System.out.print(4);
            }
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

1. package pkgA;
2. public class A {
3. private int i = 1;
4. int getI() { return i; }
5. }

B.java

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 10. Quale delle seguenti classi *non* è strettamente encapsulata?

```

class X {
    private String s;
}

class Y {

```

```

private String s;
private void setS(String s) {
    this.s = s;
}
private String getS() {
    return s;
}
}
```

```

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

#### 11. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        ((A)this).g();
    }
}
public static void main(String[] args) {
    new B().m();
}
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
-

12. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]){  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

13. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2){  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        }  
    }  
}
```

```
    } else {  
        System.out.print(0);  
    }  
}  
}  
A. 001  
B. 101  
C. 010  
D. 000  
E. 111
```

14. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;
- D. r = (Object [] []) s;
- E. r = s;

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( Exception h ) {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 j ) {  
        }  
        catch( MyExc2 x ) {  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(5);  
            throw( new MyExc2() );  
        }
```

```
    }
    catch( Exception s ) {
    }
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

- 
16. Dire quale delle seguenti affermazioni è falsa:
- 

- A. Ai metodi static non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente static e final
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 48**

1. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
}
```

```
catch( MyExc3 j ) {
}
catch( MyExc2 x ) {
}
finally {
    System.out.print(4);
    throw( new MyExc1() );
}
}
static void q() throws Exception {
try {
    System.out.print(5);
    throw( new MyExc2() );
}
catch( Exception s ) {
}
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 1524333333... (ciclo infinito)
- E. Nessuna delle precedenti

3. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

**4. Qual è l'output di questo programma?**

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

**5. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

**6. Dire quale delle seguenti affermazioni è falsa:**

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

**7. Date le dichiarazioni:**

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

8. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

9. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}
```

```
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
        a3.setRef(a1);  
        a1 = a3; a2 = a3;  
        m();  
    }  
    void m() /* do something */  
    public static void main (String[] args) {  
        new B().f();  
    }  
}  
  
A. o1  
B. o2  
C. o3  
D. o1 e o2  
E. Nessuna delle precedenti.
```

---

10. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void sets(String s) {  
        this.s = s;  
    }  
    private String gets() {  
        return s;  
    }  
}  
  
class Z {  
    private String s;  
    public void sets(String s) {  
        this.s = s;  
    }  
    public String gets() {  
        return s;  
    }  
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

11. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

15. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

---

16. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}
```

```
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 49**

1. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

2. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `s = (Error []) r;`
  - B. `r = (Object [] []) b;`
  - C. `r = b;`
  - D. `r = (Object [] []) s;`
  - E. `r = s;`
- 

3. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}
```

```
}
```

```
}
```

```
class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

4. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `u = (Integer) c;`
  - B. `d = (String) c;`
  - C. `d = (String) u;`
  - D. `u = (Integer) d;`
  - E. Nessuno dei precedenti
- 

5. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. `public void f(int j)`
- B. `public String f(byte k)`

- C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
```

```
6.     int b = new B().getI();
7.   }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
```

```

        q();
        System.out.print(2);
    }
    catch( Exception h ) {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc3 j ) {
    }
    catch( MyExc2 x ) {
    }
    finally {
        System.out.print(4);
        throw( new MyExc1() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}
}

```

- A. Errore a tempo di compilazione  
B. 1524Exception in thread main MyExc1  
C. 15  
D. 1524333333... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
    }
}

```

```

        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 341  
B. 3... (ciclo infinito)  
C. 31  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 11. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
B. public int f( int j)  
C. private void f( int j)  
D. public static void f( int j)  
E. Nessuna delle precedenti
- 

#### 12. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
B. ACD  
C. ABD

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

14. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
```

```
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.
```

---

16. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}
```

```
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 50

1. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant {  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java  
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

```
B.java  
1. package pkgB;  
2. import pkgA.A;  
3. public class B extends A {  
4.     public static void main(String[] args) {  
5.         int a = new A().getI();  
6.         int b = new B().getI();
```

```
7.     }  
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

3. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente `static` e `final`
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente `f` (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

5. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {

```

```

        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

9. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2){  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

10. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

11. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void setS(String s) {  
        this.s = s;  
    }  
    private String getS() {  
        return s;  
    }  
}  
  
class Z {  
    private String s;  
    public void setS(String s) {  
        this.s = s;  
    }  
    public String getS() {  
        return s;  
    }  
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

12. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}
```

```
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {
```

```

        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1

- C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

**14. Qual è l'output di questo programma?**

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

**15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?**

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
    }
}

```

```
a1 = a3; a2 = a3;  
m();  
}  
void m() { /* do something */ }  
public static void main (String[] args){  
    new B().f();  
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

16. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 51**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Object c;
String d;
```

```
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

3. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. o1
- B. o2
- C. o3
- D. o1 e o2

E. Nessuna delle precedenti.

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. 341

B. 3... (ciclo infinito)

C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

5. Se nella classe C è dichiarato il metodo:

```
public void f( int i )
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i )
- B. public int f( int j )
- C. private void f( int j )

D. public static void f( int j )

E. Nessuna delle precedenti

---

6. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

A. A

B. ACD

C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
    }
}
```

```

        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

8. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
- B. BB
- C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

10. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

}
if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

12. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

13. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `s = (Error []) r;`
  - B. `r = (Object [] []) b;`
  - C. `r = b;`
  - D. `r = (Object [] []) s;`
  - E. `r = s;`
- 

14. Quale delle seguenti classi *non* è strettamente encapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

15. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```

A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java
D. Errore a tempo di esecuzione alla linea 6 in B.java
E. Nessuna delle precedenti

```

---

16. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}
```

```
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 52**

1. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

2. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
    }
}
```

```
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```

---

3. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2){
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

        System.out.print(0);
    }
}

A. 001
B. 101
C. 010
D. 000
E. 111

```

---

4. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
    }
}

```

```

        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
}

```

```

A a3 = new A("o3");
void f() {
    a1.setRef(a2);
    a2.setRef(a3);
    a3.setRef(a1);
    a1 = a3; a2 = a3;
    m();
}
void m() /* do something */
public static void main (String[] args) {
    new B().f();
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

8. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

9. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente static e final
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

10. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

12. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}
```

```

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

**13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?**

```

A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

**14. Date le dichiarazioni:**

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

**15. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
```

```
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 53**

1. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
```

```
        System.out.print(2);
    }
}
static void n() {
    try {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc1 b ) {
        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}
```

- A. 341
- B. 3... (ciclo infinito)
- C. 31
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
```

```

        catch( MyExc2 x ) {
    }
    finally {
        System.out.print(4);
        throw( new MyExc1() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}
}

```

- A. Errore a tempo di compilazione  
B. 1524Exception in thread main MyExc1  
C. 15  
D. 152433333... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3  
B. Errore a tempo di compilazione  
C. 1Exception in thread main MyExc3  
D. 1... (ciclo infinito)

#### E. Nessuna delle precedenti

---

#### 5. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;  
B. d = (String) c;  
C. d = (String) u;  
D. u = (Integer) d;  
E. Nessuno dei precedenti
- 

#### 6. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001  
B. 101  
C. 010

- D. 000  
E. 111
- 

7. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente **static** e **final**
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

8. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

9. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

10. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
        a3.setRef(a1);  
        a1 = a3; a2 = a3;  
        m();  
    }  
    void m() { /* do something */ }  
    public static void main (String[] args){  
        new B().f();  
    }  
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

11. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void sets(String s) {  
        this.s = s;  
    }  
    private String gets() {  
        return s;  
    }  
}  
  
class Z {  
    private String s;  
    public void sets(String s) {  
        this.s = s;  
    }  
    public String gets() {  
        return s;  
    }  
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

12. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
}  
public static void main(String[] args) {  
    new B().m();  
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        super.g();  
        super.s = s;  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
}
```

```
Tree() {  
    super("C");  
    System.out.print("D");  
}  
}  
  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

16. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 54

1. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;
- D. r = (Object [] []) s;
- E. r = s;

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc2 g ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 c ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 14Exception in thread main MyExc3

- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

4. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        super.g();  
        super.s = s;  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

5. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

6. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
}
```

```
}
void q(String s2, Boolean b1, Boolean b2) {
    String s1;
    s1 = new String("abcde");
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
    }
}
```

```

        catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
 B. ACD  
 C. ABD  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 9. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
 B. public int f( int j)  
 C. private void f( int j)  
 D. public static void f( int j)

E. Nessuna delle precedenti

---

#### 10. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

A. A  
 B. ACD  
 C. ABD  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti

---

#### 11. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*  
 B. Un attributo può essere contemporaneamente `static` e `final`  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Quando due metodi hanno lo stesso nome non sempre si ha overloading  
 E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

#### 12. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 1524333333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

```

```

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

15. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
- B. La classe B
- C. La classe C
- D. Tutte le classi
- E. Nessuna delle classi

---

16. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
```

```
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 55**

1. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;
- D. u = (Integer) d;
- E. Nessuno dei precedenti

2. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione

- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends Error { }  
class MyExc3 extends Error { }  
public class C1 {  
    public static void main(String[] argv) {  
        try {  
            n();  
        }  
        catch( MyExc1 z ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 e ) {  
            throw( new Error() );  
        }  
        catch( MyExc3 f ) {  
            System.out.print(2);  
        }  
    }  
    static void n() {  
        try {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc1 b ) {  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 i ) {  
        }  
        catch( MyExc2 v ) {  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 341
- B. 3... (ciclo infinito)
- C. 31
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void sets(String s) {  
        this.s = s;  
    }  
    private String gets() {  
        return s;  
    }  
}  
  
class Z {  
    private String s;  
    public void sets(String s) {  
        this.s = s;  
    }  
    public String gets() {  
        return s;  
    }  
}
```

- A. La classe A
- B. La classe B
- C. La classe C
- D. Tutte le classi
- E. Nessuna delle classi

5. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

B.java

```
1. package pkgB;  
2. import pkgA.A;  
3. public class B extends A {  
4.     public static void main(String[] args) {  
5.         int a = new A().getI();  
6.         int b = new B().getI();  
7.     }  
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java

- D. Errore a tempo di esecuzione alla linea 6 in B.java

- E. Nessuna delle precedenti

6. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

7. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente static e final
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc2 g ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 c ) {  
            throw( new MyExc3() );  
        }  
    }  
}
```

```

    }
    finally {
        System.out.print(4);
    }
}

```

- A. 14Exception in thread main MyExc3  
B. Errore a tempo di compilazione  
C. 1Exception in thread main MyExc3  
D. 1... (ciclo infinito)  
E. Nessuna delle precedenti
- 

9. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
}

public static void main (String[] args) {
    new B().f();
}

```

- A. o1  
B. o2  
C. o3  
D. o1 e o2  
E. Nessuna delle precedenti.
- 

10. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;  
B. r = (Object [] []) b;  
C. r = b;  
D. r = (Object [] []) s;  
E. r = s;
- 

11. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
B. ACD  
C. ABD  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

12. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {

```

```

        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
B. BB  
C. Errore a tempo di compilazione  
D. ABException in thread main ...  
E. Nessuna delle precedenti
- 

13. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
B. public int f( int j)  
C. private void f( int j)  
D. public static void f( int j)  
E. Nessuna delle precedenti
- 

14. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {

```

```

            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

A. 001
B. 101
C. 010
D. 000
E. 111

```

---

15. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
B. BB  
C. Errore a tempo di compilazione  
D. ABException in thread main ...  
E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);

```

```
}

catch( Exception h ) {
    System.out.print(3);
    throw( new MyExc1() );
}
catch( MyExc3 j ) {
}
catch( MyExc2 x ) {
}
finally {
    System.out.print(4);
    throw( new MyExc1() );
}

static void q() throws Exception {
    try {

}
System.out.print(5);
throw( new MyExc2() );
}
catch( Exception s ) {
}
}

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 152433333... (ciclo infinito)
E. Nessuna delle precedenti
```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 56**

1. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
}
```

```
static void m() throws Exception {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
    }
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

3. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente `static` e `final`
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

4. Se nella classe C è dichiarato il metodo:

`String f(int i)`

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente `f` (overriding).

- A. `public void f(int j)`
- B. `public String f(byte k)`
- C. `protected String f(int k)`
- D. `String f(int i) throws Exception`

E. Nessuna delle precedenti.

---

5. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

6. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

```
B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

A. Nessun errore a tempo di compilazione e di esecuzione

B. Errore a tempo di compilazione

C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}
```

A. o1

B. o2

C. o3

D. o1 e o2

E. Nessuna delle precedenti.

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
    }
}
```

```

        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

9. Quale delle seguenti classi *non* è strettamente encapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A  
 B. La classe B  
 C. La classe C  
 D. Tutte le classi  
 E. Nessuna delle classi
- 

10. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
 B. BB  
 C. Errore a tempo di compilazione  
 D. ABException in thread main ...  
 E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {

```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

12. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

14. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
    }
}
```

```
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}
A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 152433333... (ciclo infinito)
E. Nessuna delle precedenti

```

---

```
catch( Exception h ) {
    System.out.print(3);
    throw( new MyExc1() );
}
catch( MyExc3 j ) {
}
catch( MyExc2 x ) {
}
finally {
    System.out.print(4);
    throw( new MyExc1() );
}
static void q() throws Exception {
    try {
        System.out.print(5);
    }
}
```



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 57**

1. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente `static` e `final`
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

3. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
}
```

```

public void g() {
    ((A)this).g();
}
public static void main(String[] args) {
    new B().m();
}
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

5. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }

```

```

public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

7. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001  
 B. 101  
 C. 010  
 D. 000  
 E. 111
- 

8. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
 B. public int f( int j)  
 C. private void f( int j)  
 D. public static void f( int j)  
 E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {

```

```

            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 1524Exception in thread main MyExc1  
 C. 15  
 D. 152433333... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
-

11. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        super.g();  
        super.s = s;  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

12. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

13. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;
- D. r = (Object [] []) s;
- E. r = s;

14. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant {  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;
- D. u = (Integer) d;
- E. Nessuno dei precedenti

16. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void setS(String s) {  
        this.s = s;  
    }  
    private String getS() {  
        return s;  
    }  
}  
  
class Z {  
    private String s;
```

```
    public void setS(String s) {  
        this.s = s;  
    }  
    public String getS() {  
        return s;  
    }  
}
```

---

- A. La classe A
- B. La classe B
- C. La classe C
- D. Tutte le classi
- E. Nessuna delle classi



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 58**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args){
        new B().f();
    }
}

A. o1
B. o2
C. o3
D. o1 e o2
E. Nessuna delle precedenti.
```

3. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
```

```

    }
}

class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
B. ACD  
C. ABD  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

1. package pkgA;
2. public class A {
3. private int i = 1;
4. int getI() { return i; }
5. }

B.java

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4. public static void main(String[] args) {
5. int a = new A().getI();
6. int b = new B().getI();
7. }
8. }

- A. Nessun errore a tempo di compilazione e di esecuzione  
B. Errore a tempo di compilazione  
C. Errore a tempo di esecuzione alla linea 5 in B.java  
D. Errore a tempo di esecuzione alla linea 6 in B.java  
E. Nessuna delle precedenti
- 

5. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{

```

```

Tree() {
    super("C");
    System.out.print("D");
}
}

class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
B. ACD  
C. ABD  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

6. Se nella classe C è dichiarato il metodo:

String f(int i)

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)  
B. public String f(byte k)  
C. protected String f(int k)  
D. String f(int i) throws Exception  
E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

---

8. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

---

9. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void setS(String s) {  
        this.s = s;  
    }  
    private String getS() {  
        return s;  
    }  
}  
  
class Z {  
    private String s;  
    public void setS(String s) {  
        this.s = s;  
    }  
    public String getS() {  
        return s;  
    }  
}
```

- A. La classe A
- B. La classe B
- C. La classe C
- D. Tutte le classi
- E. Nessuna delle classi

---

10. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente static e final
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends Error { }  
class MyExc3 extends Error { }  
public class C1 {  
    public static void main(String[] argv) {  
        try {  
            n();  
        }  
        catch( MyExc1 z ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 e ) {  
            throw( new Error() );  
        }  
        catch( MyExc3 f ) {  
            System.out.print(2);  
        }  
    }  
    static void n() {  
        try {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc1 b ) {  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 i ) {  
        }  
        catch( MyExc2 v ) {  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 341
- B. 3... (ciclo infinito)
- C. 31
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2) {  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

13. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
    }  
}
```

```
    g();  
}  
public void g() {  
    super.g();  
    super.s = s;  
}  
public static void main(String[] args) {  
    new B().m();  
}
```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc2 g ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 c ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

15. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

16. Date le dichiarazioni:

```
Object c;  
String d;
```

---

```
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 59

1. Qual è l'output di questo programma?

```
class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
  
class Tree extends Plant{  
    Tree(String s) {  
        super("C");  
        System.out.print(s);  
    }  
}  
  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

3. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente `static` e `final`
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

4. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `s = (Error []) r;`
- B. `r = (Object [] []) b;`
- C. `r = b;`
- D. `r = (Object [] []) s;`
- E. `r = s;`

5. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void sets(String s) {  
        this.s = s;  
    }  
    private String gets() {  
        return s;  
    }  
}
```

```
class Z {
```

```

private String s;
public void sets(String s) {
    this.s = s;
}
public String gets() {
    return s;
}
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

6. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {

```

```

            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

```

        throw( new MyExc1() );
    }
}

static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 1524333333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

10. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
- B. 101
- C. 010

D. 000

E. 111

---

11. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
}
public static void main(String[] args) {
}

```

```

        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```

1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

B.java

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
}

```

```

        }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
        a3.setRef(a1);  
    }  
    a1 = a3; a2 = a3;  
    m();  
}  
void m() { /* do something */ }  
public static void main (String[] args) {  
    new B().f();  
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 60**

1. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
```

```
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

3. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
}
```

```

void g() {}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

5. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente **static** e **final**
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di **default**

---

6. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```

A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

A. Nessun errore a tempo di compilazione e di esecuzione
B. Errore a tempo di compilazione
C. Errore a tempo di esecuzione alla linea 5 in B.java

```

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

8. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}
```

```
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
        a3.setRef(a1);  
        a1 = a3; a2 = a3;  
        m();  
    }  
    void m() { /* do something */ }  
    public static void main (String[] args){  
        new B().f();  
    }  
}
```

A. o1

B. o2

C. o3

D. o1 e o2

E. Nessuna delle precedenti.

---

9. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(
```

```
                super.s + s);  
    }  
    public void g() {  
        super.g();  
        super.s = s;  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc2 g ) {  
            System.out.print(3);  
        }  
        }  
        static void m() throws Exception {  
        try {  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 c ) {  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

A. 14Exception in thread main MyExc3

B. Errore a tempo di compilazione

C. 1Exception in thread main MyExc3

D. 1... (ciclo infinito)

E. Nessuna delle precedenti

---

11. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;
- D. r = (Object [] []) s;
- E. r = s;

13. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void sets(String s) {  
        this.s = s;  
    }  
    private String gets() {  
        return s;  
    }  
}  
  
class Z {  
    private String s;  
    public void sets(String s) {  
        this.s = s;  
    }  
    public String gets() {  
        return s;  
    }  
}
```

- A. La classe A
- B. La classe B
- C. La classe C
- D. Tutte le classi

E. Nessuna delle classi

14. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]){  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}  
  
A. A  
B. ACD  
C. ABD  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
```

15. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends Error { }  
class MyExc3 extends Error { }  
public class C1 {  
    public static void main(String[] argv) {  
        try {  
            n();  
        }  
        catch( MyExc1 z ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 e ) {  
            throw( new Error() );  
        }  
        catch( MyExc3 f ) {  
            System.out.print(2);  
        }  
    }  
    static void n() {  
        try {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc1 b ) {  
            throw( new MyExc1() );  
        }  
    }  
}
```

```
}

catch( MyExc3 i ) {
}

catch( MyExc2 v ) {
}

finally {
    System.out.print(4);
}

}
```

- A. 341
- B. 3... (ciclo infinito)
- C. 31
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

16. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 61

1. Date le dichiarazioni:

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;
- D. u = (Integer) d;
- E. Nessuno dei precedenti

2. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( Exception h ) {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 j ) {  
        }  
        catch( MyExc2 x ) {  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(5);  
            throw( new MyExc2() );  
        }  
        catch( Exception s ) {  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

5. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

```

```

abstract class A implements I {
    String s = "A";
    void g() {}
}

```

```

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
- B. ACD
- C. ABD

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
}

```

```

public static void main(String argv[]) {
    System.out.print("A");
    Palm p = new Palm();
}

```

- A. A  
 B. ACD  
 C. ABD  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

9. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}

```

- A. La classe A  
 B. La classe B  
 C. La classe C  
 D. Tutte le classi  
 E. Nessuna delle classi
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
    }
}

```

```

catch( MyExc1 z ) {
    System.out.print(1);
}
catch( MyExc2 e ) {
    throw( new Error() );
}
catch( MyExc3 f ) {
    System.out.print(2);
}
static void n() {
    try {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc1 b ) {
        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```

1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

B.java

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione  
 B. Errore a tempo di compilazione  
 C. Errore a tempo di esecuzione alla linea 5 in B.java

D. Errore a tempo di esecuzione alla linea 6 in B.java

E. Nessuna delle precedenti

---

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
        a3.setRef(a1);  
        a1 = a3; a2 = a3;  
        m();  
    }  
    void m() { /* do something */ }  
    public static void main (String[] args){  
        new B().f();  
    }  
}
```

A. o1

B. o2

C. o3

D. o1 e o2

E. Nessuna delle precedenti.

---

13. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2){  
        String s1;
```

```
s1 = new String("abcde");  
if(s2 == s1) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}  
if(b1 == b2) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}  
if(s4 == s5) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}
```

A. 001

B. 101

C. 010

D. 000

E. 111

---

14. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;
- D. r = (Object [] []) s;
- E. r = s;

---

15. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi **static** non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente **static** e **final**
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

16. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
}  
  
}  
  
public void g() {  
    super.g();  
    super.s = s;  
}  
public static void main(String[] args) {  
    new B().m();  
}
```

---

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 62**

1. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

2. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
```

```
A ref;
static Object o;
A(Object o) {
    this.o = o;
}
void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
}
public static void main (String[] args) {
    new B().f();
}
```

- A. o1
- B. o2
- C. o3
- D. o1 e o2
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

B.java
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
```

```
5.     int a = new A().getI();
6.     int b = new B().getI();
7. }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

#### 4. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

#### 5. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)

- B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

#### 6. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente static e final
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

#### 7. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 1524Exception in thread main MyExc1  
C. 15  
D. 152433333... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
}

```

```

        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341  
B. 3... (ciclo infinito)  
C. 31  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 10. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;  
B. r = (Object [] []) b;  
C. r = b;  
D. r = (Object [] []) s;  
E. r = s;
- 

#### 11. Quale delle seguenti classi *non* è strettamente encapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

```

```

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

12. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

14. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

15. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
-

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
```

```
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 63**

1. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

4. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
```

```

    }
}

class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

7. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

9. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
    void q(String s2, Boolean b1, Boolean b2){  
        String s1;  
        s1 = new String("abcde");  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

10. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {  
    private String s;  
}  
  
class Y {  
    private String s;  
    private void setS(String s) {  
        this.s = s;  
    }  
    private String getS() {  
        return s;  
    }  
}
```

```
class Z {  
    private String s;  
    public void setS(String s) {  
        this.s = s;  
    }  
    public String getS() {  
        return s;  
    }  
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

11. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java  
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

```
B.java  
1. package pkgB;  
2. import pkgA.A;  
3. public class B extends A {  
4.     public static void main(String[] args){  
5.         int a = new A().getI();  
6.         int b = new B().getI();  
7.     }  
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java

- E. Nessuna delle precedenti

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. 341

B. 3... (ciclo infinito)

C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

13. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente `static` e `final`
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri

- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading

- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

14. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

A. AB

B. BB

C. Errore a tempo di compilazione

D. ABException in thread main ...

E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
}
```

```
}

static void m() throws Exception {
    try {
        throw( new MyExc3() );
    }
    catch( MyExc3 c ) {
        throw( new MyExc3() );
    }
    finally {
        System.out.print(4);
    }
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 64**

1. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

2. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
```

```
class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
}
```

```

    }
}

static void n() {
    try {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc1 b ) {
        throw( new MyExc1() );
    }
    catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 341  
B. 3... (ciclo infinito)  
C. 31  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)  
B. public String f(byte k)  
C. protected String f(int k)  
D. String f(int i) throws Exception  
E. Nessuna delle precedenti.
- 

5. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;  
B. r = (Object [] []) b;  
C. r = b;  
D. r = (Object [] []) s;  
E. r = s;
- 

6. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;  
B. d = (String) c;  
C. d = (String) u;  
D. u = (Integer) d;  
E. Nessuno dei precedenti
- 

7. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB  
B. BB  
C. Errore a tempo di compilazione  
D. ABException in thread main ...  
E. Nessuna delle precedenti
- 

8. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
 B. ACD  
 C. ABD  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

9. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
 B. public int f( int j)  
 C. private void f( int j)  
 D. public static void f( int j)  
 E. Nessuna delle precedenti
- 

10. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {

```

```

        System.out.print(
            super.s + s);
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
 B. BB  
 C. Errore a tempo di compilazione  
 D. ABException in thread main ...  
 E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```

1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione  
 B. Errore a tempo di compilazione  
 C. Errore a tempo di esecuzione alla linea 5 in B.java  
 D. Errore a tempo di esecuzione alla linea 6 in B.java  
 E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);

```

```

        q();
        System.out.print(2);
    }
    catch( Exception h ) {
        System.out.print(3);
        throw( new MyExc1() );
    }
    catch( MyExc3 j ) {
    }
    catch( MyExc2 x ) {
    }
    finally {
        System.out.print(4);
        throw( new MyExc1() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}
}

```

- A. Errore a tempo di compilazione  
B. 1524Exception in thread main MyExc1  
C. 15  
D. 1524333333... (ciclo infinito)  
E. Nessuna delle precedenti
- 

### 13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- ```

        }
    }
}

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

```
- 

### 14. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args){
        new B().f();
    }
}

```

- A. o1  
B. o2  
C. o3  
D. o1 e o2  
E. Nessuna delle precedenti.
- 

### 15. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{

```

```
Tree() {  
    super("C");  
    System.out.print("D");  
}  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

16. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 65**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

4. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

6. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

7. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
A
ACD
ABD
Errore a tempo di compilazione
Nessuna delle precedenti
```

---

9. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;

- D. `u = (Integer) d;`  
 E. Nessuno dei precedenti
- 

10. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 001  
 B. 101  
 C. 010  
 D. 000  
 E. 111
- 

11. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}
```

```
class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() /* do something */ {
        public static void main (String[] args) {
            new B().f();
        }
    }
}
```

- A. o1  
 B. o2  
 C. o3  
 D. o1 e o2  
 E. Nessuna delle precedenti.
- 

12. Quale delle seguenti classi *non* è strettamente incapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A  
 B. La classe B  
 C. La classe C  
 D. Tutte le classi  
 E. Nessuna delle classi
-

13. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
public class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        super.g();  
        super.s = s;  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
            System.out.print(2);  
        }  
        catch( MyExc2 g ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 c ) {  
            throw( new MyExc3() );  
        }  
    }  
}
```

```
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

15. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }
```

```
}

catch( Exception h ) {
    System.out.print(3);
    throw( new MyExc1() );
}
catch( MyExc3 j ) {
}
catch( MyExc2 x ) {
}
finally {
    System.out.print(4);
    throw( new MyExc1() );
}

static void q() throws Exception {
    try {

}
System.out.print(5);
throw( new MyExc2() );
}
catch( Exception s ) {
}
}

A. Errore a tempo di compilazione
B. 1524Exception in thread main MyExc1
C. 15
D. 152433333... (ciclo infinito)
E. Nessuna delle precedenti
```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 66**

1. Quale delle seguenti classi *non* è strettamente encapsulata?

```
class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}
```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

2. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
- B. r = (Object [] []) b;
- C. r = b;

- D. r = (Object [] []) s;
  - E. r = s;
- 

3. Qual è l'output di questo programma?

```
class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi *static* non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente *static* e *final*
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

---

5. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. 341

B. 3... (ciclo infinito)

C. 31

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

7. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}
```

A. o1

B. o2

C. o3

D. o1 e o2

E. Nessuna delle precedenti.

---

8. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
```

```

class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

```

```

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

1. package pkgA;
2. public class A {
3. private int i = 1;
4. int getI() { return i; }
5. }

B.java

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4. public static void main(String[] args) {
5. int a = new A().getI();
6. int b = new B().getI();
7. }
8. }

- A. Nessun errore a tempo di compilazione e di esecuzione
- B. Errore a tempo di compilazione
- C. Errore a tempo di esecuzione alla linea 5 in B.java
- D. Errore a tempo di esecuzione alla linea 6 in B.java
- E. Nessuna delle precedenti

---

12. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
- B. public String f(byte k)
- C. protected String f(int k)
- D. String f(int i) throws Exception
- E. Nessuna delle precedenti.

---

### 13. Qual è l'output di questo programma?

```
interface I {  
    String s = "I";  
    void m();  
}  
  
abstract class A implements I {  
    String s = "A";  
    void g() {}  
}  
  
class B extends A {  
    String s = "B";  
    public void m() {  
        System.out.print(  
            super.s + s);  
        g();  
    }  
    public void g() {  
        ((A)this).g();  
    }  
    public static void main(String[] args) {  
        new B().m();  
    }  
}
```

- A. AB
- B. BB
- C. Errore a tempo di compilazione
- D. ABException in thread main ...
- E. Nessuna delle precedenti

---

### 14. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)

E. Nessuna delle precedenti

---

### 15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( Exception h ) {  
            System.out.print(3);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 j ) {  
        }  
        catch( MyExc2 x ) {  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(5);  
            throw( new MyExc2() );  
        }  
        catch( Exception s ) {  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 1524Exception in thread main MyExc1
- C. 15
- D. 152433333... (ciclo infinito)
- E. Nessuna delle precedenti

---

### 16. Quale output si ottiene invocando il metodo p?

```
class C {  
    private Boolean b3 = new Boolean(true);  
    private Boolean b4;  
    private String s4 = new String("ab");  
    private String s5;  
    void p() {  
        String s3;  
        s3 = new String("abcde");  
        b4 = b3;  
        s5 = new String("ab");  
        q(s3, b3, b4);  
    }  
}
```

```
void q(String s2, Boolean b1, Boolean b2){  
    String s1;  
    s1 = new String("abcde");  
    if(s2 == s1) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(b1 == b2) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(s4 == s5) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
}
```

---

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 67**

1. Date le dichiarazioni:

```
Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

2. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
```

```
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
    finally {
        System.out.print(4);
        throw( new MyExc1() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(5);
        throw( new MyExc2() );
    }
    catch( Exception s ) {
    }
}
```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 1524333333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo programma?

```
abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
```

```

public static void main(String argv[]) {
    System.out.print("A");
    Palm p = new Palm();
}
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void sets(String s) {
        this.s = s;
    }
    public String gets() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

6. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

```

```

    }

    public class B extends A {
        String s = "B";
        public void m() {
            System.out.print(
                super.s + s);
        }
        public void g() {
            super.g();
            super.s = s;
        }
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

8. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
}

```

```

public void g() {
    ((A)this).g();
}
public static void main(String[] args) {
    new B().m();
}
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

9. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001
  - B. 101
  - C. 010
  - D. 000
  - E. 111
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 14Exception in thread main MyExc3
  - B. Errore a tempo di compilazione
  - C. 1Exception in thread main MyExc3
  - D. 1... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A

- B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] args) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 341
  - B. 3... (ciclo infinito)
  - C. 31
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Se nella classe C è dichiarato il metodo:

```
public void f( int i )
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i )
  - B. public int f( int j )
  - C. private void f( int j )
  - D. public static void f( int j )
  - E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

1. package pkgA;
2. public class A {
3. private int i = 1;
4. int getI() { return i; }
5. }

B.java

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4. public static void main(String[] args) {
5. int a = new A().getI();
6. int b = new B().getI();
7. }
8. }

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```
Object [] b;
Object [] [] r;
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

16. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {  
    A ref;  
    static Object o;  
    A(Object o) {  
        this.o = o;  
    }  
    void setRef(A a) {ref = a;}  
}  
  
class B {  
    A a1 = new A("o1");  
    A a2 = new A("o2");  
    A a3 = new A("o3");  
    void f() {  
        a1.setRef(a2);  
        a2.setRef(a3);  
        a3.setRef(a1);  
    }  
    a1 = a3; a2 = a3;  
    m();  
}  
void m() { /* do something */ }  
public static void main (String[] args) {  
    new B().f();  
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 68

1. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
- B. Un attributo può essere contemporaneamente `static` e `final`
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
- E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

2. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `s = (Error []) r;`
- B. `r = (Object [] []) b;`
- C. `r = b;`
- D. `r = (Object [] []) s;`
- E. `r = s;`

3. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. `public void f(int j)`
- B. `public String f(byte k)`
- C. `protected String f(int k)`
- D. `String f(int i) throws Exception`
- E. Nessuna delle precedenti.

4. Qual è l'output di questo programma?

```
abstract class Plant {  
    Plant() {  
        System.out.print("B");  
    }  
    Plant(String s) {  
        System.out.print(s);  
    }  
}  
class Tree extends Plant{  
    Tree() {  
        super("C");  
        System.out.print("D");  
    }  
}  
class Palm extends Tree {  
    public static void main(String argv[]) {  
        System.out.print("A");  
        Palm p = new Palm();  
    }  
}
```

- A. A
- B. ACD
- C. ABD
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java  
1. package pkgA;  
2. public class A {  
3.     private int i = 1;  
4.     int getI() { return i; }  
5. }
```

```
B.java  
1. package pkgB;  
2. import pkgA.A;  
3. public class B extends A {  
4.     public static void main(String[] args) {
```

```

5.     int a = new A().getI();
6.     int b = new B().getI();
7. }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}
```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}
```

```

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}
```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

#### 9. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

```

```

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {

```

```

            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
    A. Errore a tempo di compilazione
    B. 1524Exception in thread main MyExc1
    C. 15
    D. 152433333... (ciclo infinito)
    E. Nessuna delle precedenti

```

---

#### 11. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

#### 12. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {

```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 001  
B. 101  
C. 010  
D. 000  
E. 111
- 

13. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;  
B. d = (String) c;  
C. d = (String) u;  
D. u = (Integer) d;  
E. Nessuno dei precedenti
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
    }
}

```

```

        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341  
B. 3... (ciclo infinito)  
C. 31  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

15. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
    public static void main (String[] args) {
        new B().f();
    }
}

```

- A. o1  
B. o2  
C. o3  
D. o1 e o2  
E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );

```

- ```

        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}

A. 14Exception in thread main MyExc3
B. Errore a tempo di compilazione
C. 1Exception in thread main MyExc3
D. 1... (ciclo infinito)
E. Nessuna delle precedenti

```
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 69**

1. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```
class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
}
public static void main (String[] args) {
    new B().f();
}
```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

```
A.java
1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }
```

B.java

```
1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }
```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
    }
}
```

```

    }
    catch( MyExc3 i ) {
    }
    catch( MyExc2 v ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 341  
B. 3... (ciclo infinito)  
C. 31  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

4. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)  
B. public String f(byte k)  
C. protected String f(int k)  
D. String f(int i) throws Exception  
E. Nessuna delle precedenti.
- 

5. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi static non si applica il *dynamic method dispatch*  
B. Un attributo può essere contemporaneamente static e final  
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
D. Quando due metodi hanno lo stesso nome non sempre si ha overloading  
E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

6. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {

```

```

    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
    }
    public void g() {
        ((A)this).g();
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB  
B. BB  
C. Errore a tempo di compilazione  
D. ABException in thread main ...  
E. Nessuna delle precedenti
- 

7. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A  
B. ACD  
C. ABD  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

8. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree() {
        super("C");
        System.out.print("D");
    }
}
class Palm extends Tree {
    public static void main(String argv[]){
        System.out.print("A");
        Palm p = new Palm();
    }
}

```

- A. A
  - B. ACD
  - C. ABD
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo programma?

```

interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
- B. BB

- C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

#### 10. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void setS(String s) {
        this.s = s;
    }
    private String getS() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

#### 11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
    }
}

```

```

        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 1524Exception in thread main MyExc1  
C. 15  
D. 152433333... (ciclo infinito)  
E. Nessuna delle precedenti
- 

12. Se nella classe C è dichiarato il metodo p:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
B. public int f( int j)  
C. private void f( int j)  
D. public static void f( int j)  
E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }

```

```

        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 14Exception in thread main MyExc3  
B. Errore a tempo di compilazione  
C. 1Exception in thread main MyExc3  
D. 1... (ciclo infinito)  
E. Nessuna delle precedenti
- 

14. Quale output si ottiene invocando il metodo p?

```

class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
        b4 = b3;
        s5 = new String("ab");
        q(s3, b3, b4);
    }
    void q(String s2, Boolean b1, Boolean b2) {
        String s1;
        s1 = new String("abcde");
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 001  
B. 101  
C. 010  
D. 000  
E. 111
- 

15. Date le dichiarazioni:

```
Object [] b;  
Object [] [] r;  
Error [] s;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

16. Date le dichiarazioni:

---

```
Object c;  
String d;  
Integer u;  
c = new Integer(0);  
d = new String("abcd");  
u = new Integer(1);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
- B. d = (String) c;
- C. d = (String) u;
- D. u = (Integer) d;
- E. Nessuno dei precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 70**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv)
                    throws Exception {
        try {
            System.out.print(1);
            m();
            System.out.print(2);
        }
        catch( MyExc2 g ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc3() );
        }
        catch( MyExc3 c ) {
            throw( new MyExc3() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 14Exception in thread main MyExc3
- B. Errore a tempo di compilazione
- C. 1Exception in thread main MyExc3
- D. 1... (ciclo infinito)
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo p?

```
class C {
    private Boolean b3 = new Boolean(true);
    private Boolean b4;
    private String s4 = new String("ab");
    private String s5;
    void p() {
        String s3;
        s3 = new String("abcde");
```

```
b4 = b3;
s5 = new String("ab");
q(s3, b3, b4);
}
void q(String s2, Boolean b1, Boolean b2) {
    String s1;
    s1 = new String("abcde");
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 001
- B. 101
- C. 010
- D. 000
- E. 111

3. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

public class B extends A {
    String s = "B";
    public void m() {
```

```

        System.out.print(
            super.s + s);
        g();
    }
    public void g() {
        super.g();
        super.s = s;
    }
    public static void main(String[] args) {
        new B().m();
    }
}

```

- A. AB
  - B. BB
  - C. Errore a tempo di compilazione
  - D. ABException in thread main ...
  - E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```

Object c;
String d;
Integer u;
c = new Integer(0);
d = new String("abcd");
u = new Integer(1);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. u = (Integer) c;
  - B. d = (String) c;
  - C. d = (String) u;
  - D. u = (Integer) d;
  - E. Nessuno dei precedenti
- 

5. Se nella classe C è dichiarato il metodo:

```
String f(int i)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(int j)
  - B. public String f(byte k)
  - C. protected String f(int k)
  - D. String f(int i) throws Exception
  - E. Nessuna delle precedenti.
- 

6. Quale delle seguenti classi *non* è strettamente incapsulata?

```

class X {
    private String s;
}

class Y {
    private String s;
    private void sets(String s) {
        this.s = s;
    }
    private String gets() {
        return s;
    }
}

class Z {
    private String s;
    public void setS(String s) {
        this.s = s;
    }
    public String getS() {
        return s;
    }
}

```

- A. La classe A
  - B. La classe B
  - C. La classe C
  - D. Tutte le classi
  - E. Nessuna delle classi
- 

7. Date le dichiarazioni:

```

Object [] b;
Object [] [] r;
Error [] s;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. s = (Error []) r;
  - B. r = (Object [] []) b;
  - C. r = b;
  - D. r = (Object [] []) s;
  - E. r = s;
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends Error { }
class MyExc3 extends Error { }
public class C1 {
    public static void main(String[] argv) {
        try {
            n();
        }
        catch( MyExc1 z ) {
            System.out.print(1);
        }
    }
}

```

```

        catch( MyExc2 e ) {
            throw( new Error() );
        }
        catch( MyExc3 f ) {
            System.out.print(2);
        }
    }
    static void n() {
        try {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc1 b ) {
            throw( new MyExc1() );
        }
        catch( MyExc3 i ) {
        }
        catch( MyExc2 v ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 341  
 B. 3... (ciclo infinito)  
 C. 31  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

9. Se nella classe C è dichiarato il metodo:

```
public void f( int i)
```

indicare quale delle seguenti ulteriori dichiarazioni può essere correttamente inserita in C.

- A. private void f( float i)  
 B. public int f( int j)  
 C. private void f( int j)  
 D. public static void f( int j)  
 E. Nessuna delle precedenti
- 

10. Qual è l'output di questo programma?

```

abstract class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{

```

```

Tree() {
    super("C");
    System.out.print("D");
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

11. Qual è l'output di questo programma?

```

class Plant {
    Plant() {
        System.out.print("B");
    }
    Plant(String s) {
        System.out.print(s);
    }
}
class Tree extends Plant{
    Tree(String s) {
        super("C");
        System.out.print(s);
    }
}
class Palm extends Tree {
    public static void main(String argv[]) {
        System.out.print("A");
        Palm p = new Palm();
    }
}
A. A
B. ACD
C. ABD
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

12. Quali dei tre oggetti a1, a2 o a3 sono eleggibili per la garbage collection quando il metodo m comincia ad essere eseguito?

```

class A {
    A ref;
    static Object o;
    A(Object o) {
        this.o = o;
    }
    void setRef(A a) {ref = a;}
}

```

```

}

class B {
    A a1 = new A("o1");
    A a2 = new A("o2");
    A a3 = new A("o3");
    void f() {
        a1.setRef(a2);
        a2.setRef(a3);
        a3.setRef(a1);
        a1 = a3; a2 = a3;
        m();
    }
    void m() { /* do something */ }
}
public static void main (String[] args) {
    new B().f();
}
}

```

- A. o1
  - B. o2
  - C. o3
  - D. o1 e o2
  - E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione dei seguenti file?

A.java

```

1. package pkgA;
2. public class A {
3.     private int i = 1;
4.     int getI() { return i; }
5. }

```

B.java

```

1. package pkgB;
2. import pkgA.A;
3. public class B extends A {
4.     public static void main(String[] args) {
5.         int a = new A().getI();
6.         int b = new B().getI();
7.     }
8. }

```

- A. Nessun errore a tempo di compilazione e di esecuzione
  - B. Errore a tempo di compilazione
  - C. Errore a tempo di esecuzione alla linea 5 in B.java
  - D. Errore a tempo di esecuzione alla linea 6 in B.java
  - E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Ai metodi `static` non si applica il *dynamic method dispatch*
  - B. Un attributo può essere contemporaneamente `static` e `final`
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Quando due metodi hanno lo stesso nome non sempre si ha overloading
  - E. Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
            throw( new MyExc1() );
        }
        catch( MyExc3 j ) {
        }
        catch( MyExc2 x ) {
        }
        finally {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(5);
            throw( new MyExc2() );
        }
        catch( Exception s ) {
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 1524Exception in thread main MyExc1
  - C. 15
  - D. 152433333... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo programma?

```
interface I {
    String s = "I";
    void m();
}

abstract class A implements I {
    String s = "A";
    void g() {}
}

class B extends A {
    String s = "B";
    public void m() {
        System.out.print(
            super.s + s);
        g();
    }
}

public void g() {
    ((A)this).g();
}
public static void main(String[] args) {
    new B().m();
}
```

---

A. AB  
B. BB  
C. Errore a tempo di compilazione  
D. ABException in thread main ...  
E. Nessuna delle precedenti

**Prova n. 1**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 2**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 3**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 4**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 5**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 6**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 8**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 9**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 10**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 11**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 12**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 13**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 15**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 16**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

14 giugno 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

## Risultato prova n. 1:

A 16x16 grid containing red squares at the following coordinates:

- (2, 2)
- (3, 3)
- (4, 4)
- (5, 5)
- (6, 6)
- (7, 7)
- (8, 8)
- (9, 9)
- (10, 10)
- (11, 11)
- (12, 12)
- (13, 13)
- (14, 14)
- (15, 15)
- (16, 16)
- (1, 16)

## Risultato prova n. 2:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 3:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  |   |   |   |   |   | ■ | ■ |
| 5  | ■ | ■ |   |   |   |   |   |
| 6  |   |   |   | ■ | ■ |   |   |
| 7  |   |   | ■ | ■ |   |   |   |
| 8  | ■ | ■ |   |   |   |   |   |
| 9  |   |   |   |   |   | ■ | ■ |
| 10 | ■ | ■ |   |   |   |   |   |
| 11 | ■ | ■ |   |   |   |   |   |
| 12 |   |   |   | ■ | ■ |   |   |
| 13 |   |   |   | ■ | ■ |   |   |
| 14 | ■ | ■ |   |   |   |   |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 |   |   |   | ■ | ■ |   |   |

Risultato prova n. 4:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 5:

|    |  |   |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|---|
| 1  |  |   | ■ | ■ |   |   |   |   |
| 2  |  |   |   | ■ | ■ |   |   |   |
| 3  |  | ■ | ■ |   |   |   |   |   |
| 4  |  |   |   |   |   | ■ | ■ |   |
| 5  |  |   |   |   | ■ | ■ |   |   |
| 6  |  |   |   |   |   |   | ■ | ■ |
| 7  |  | ■ | ■ |   |   |   |   |   |
| 8  |  |   |   |   |   | ■ | ■ |   |
| 9  |  | ■ | ■ |   |   |   |   |   |
| 10 |  |   |   |   |   |   | ■ | ■ |
| 11 |  | ■ | ■ |   |   |   |   |   |
| 12 |  | ■ | ■ |   |   |   |   |   |
| 13 |  |   |   |   | ■ | ■ |   |   |
| 14 |  |   |   |   | ■ | ■ |   |   |
| 15 |  |   |   | ■ | ■ |   |   |   |
| 16 |  | ■ | ■ |   |   |   |   |   |

Risultato prova n. 6:

A 16x16 grid containing red squares at the following coordinates:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (1, 5)
- (2, 5)
- (3, 5)
- (4, 5)
- (5, 5)
- (6, 5)
- (7, 5)
- (8, 5)
- (9, 5)
- (10, 5)
- (11, 5)
- (12, 5)
- (13, 5)
- (14, 5)
- (15, 5)
- (16, 5)
- (1, 6)
- (2, 6)
- (3, 6)
- (4, 6)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (1, 7)
- (2, 7)
- (3, 7)
- (4, 7)
- (5, 7)
- (6, 7)
- (7, 7)
- (8, 7)
- (9, 7)
- (10, 7)
- (11, 7)
- (12, 7)
- (13, 7)
- (14, 7)
- (15, 7)
- (16, 7)
- (1, 8)
- (2, 8)
- (3, 8)
- (4, 8)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (1, 9)
- (2, 9)
- (3, 9)
- (4, 9)
- (5, 9)
- (6, 9)
- (7, 9)
- (8, 9)
- (9, 9)
- (10, 9)
- (11, 9)
- (12, 9)
- (13, 9)
- (14, 9)
- (15, 9)
- (16, 9)
- (1, 10)
- (2, 10)
- (3, 10)
- (4, 10)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (1, 11)
- (2, 11)
- (3, 11)
- (4, 11)
- (5, 11)
- (6, 11)
- (7, 11)
- (8, 11)
- (9, 11)
- (10, 11)
- (11, 11)
- (12, 11)
- (13, 11)
- (14, 11)
- (15, 11)
- (16, 11)
- (1, 12)
- (2, 12)
- (3, 12)
- (4, 12)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (1, 13)
- (2, 13)
- (3, 13)
- (4, 13)
- (5, 13)
- (6, 13)
- (7, 13)
- (8, 13)
- (9, 13)
- (10, 13)
- (11, 13)
- (12, 13)
- (13, 13)
- (14, 13)
- (15, 13)
- (16, 13)
- (1, 14)
- (2, 14)
- (3, 14)
- (4, 14)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)
- (1, 15)
- (2, 15)
- (3, 15)
- (4, 15)
- (5, 15)
- (6, 15)
- (7, 15)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 15)
- (16, 15)
- (1, 16)
- (2, 16)
- (3, 16)
- (4, 16)
- (5, 16)
- (6, 16)
- (7, 16)
- (8, 16)
- (9, 16)
- (10, 16)
- (11, 16)
- (12, 16)
- (13, 16)
- (14, 16)
- (15, 16)
- (16, 16)

## Risultato prova n. 7:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 8:

A 16x16 grid with red squares at the following coordinates:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (1, 5)
- (2, 5)
- (3, 5)
- (4, 5)
- (5, 5)
- (6, 5)
- (7, 5)
- (8, 5)
- (9, 5)
- (10, 5)
- (11, 5)
- (12, 5)
- (13, 5)
- (14, 5)
- (15, 5)
- (16, 5)
- (1, 6)
- (2, 6)
- (3, 6)
- (4, 6)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (1, 7)
- (2, 7)
- (3, 7)
- (4, 7)
- (5, 7)
- (6, 7)
- (7, 7)
- (8, 7)
- (9, 7)
- (10, 7)
- (11, 7)
- (12, 7)
- (13, 7)
- (14, 7)
- (15, 7)
- (16, 7)
- (1, 8)
- (2, 8)
- (3, 8)
- (4, 8)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (1, 9)
- (2, 9)
- (3, 9)
- (4, 9)
- (5, 9)
- (6, 9)
- (7, 9)
- (8, 9)
- (9, 9)
- (10, 9)
- (11, 9)
- (12, 9)
- (13, 9)
- (14, 9)
- (15, 9)
- (16, 9)
- (1, 10)
- (2, 10)
- (3, 10)
- (4, 10)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (1, 11)
- (2, 11)
- (3, 11)
- (4, 11)
- (5, 11)
- (6, 11)
- (7, 11)
- (8, 11)
- (9, 11)
- (10, 11)
- (11, 11)
- (12, 11)
- (13, 11)
- (14, 11)
- (15, 11)
- (16, 11)
- (1, 12)
- (2, 12)
- (3, 12)
- (4, 12)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (1, 13)
- (2, 13)
- (3, 13)
- (4, 13)
- (5, 13)
- (6, 13)
- (7, 13)
- (8, 13)
- (9, 13)
- (10, 13)
- (11, 13)
- (12, 13)
- (13, 13)
- (14, 13)
- (15, 13)
- (16, 13)
- (1, 14)
- (2, 14)
- (3, 14)
- (4, 14)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)
- (1, 15)
- (2, 15)
- (3, 15)
- (4, 15)
- (5, 15)
- (6, 15)
- (7, 15)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 15)
- (16, 15)
- (1, 16)
- (2, 16)
- (3, 16)
- (4, 16)
- (5, 16)
- (6, 16)
- (7, 16)
- (8, 16)
- (9, 16)
- (10, 16)
- (11, 16)
- (12, 16)
- (13, 16)
- (14, 16)
- (15, 16)
- (16, 16)

## Risultato prova n. 9:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 10:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 11:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 12:

|    |   |  |   |   |   |   |  |
|----|---|--|---|---|---|---|--|
| 1  |   |  | ■ |   |   |   |  |
| 2  | ■ |  |   |   |   |   |  |
| 3  |   |  |   | ■ |   |   |  |
| 4  | ■ |  |   |   |   |   |  |
| 5  |   |  |   |   | ■ |   |  |
| 6  |   |  |   |   |   | ■ |  |
| 7  |   |  | ■ |   |   |   |  |
| 8  | ■ |  |   |   |   |   |  |
| 9  |   |  |   |   | ■ |   |  |
| 10 | ■ |  |   |   |   |   |  |
| 11 | ■ |  |   |   |   |   |  |
| 12 |   |  |   | ■ |   |   |  |
| 13 |   |  |   | ■ | ■ |   |  |
| 14 |   |  | ■ |   |   |   |  |
| 15 | ■ |  |   |   |   |   |  |
| 16 |   |  |   |   |   | ■ |  |

Risultato prova n. 13:

|    |   |   |   |   |   |  |   |
|----|---|---|---|---|---|--|---|
| 1  |   | ■ |   |   |   |  |   |
| 2  |   |   |   |   |   |  | ■ |
| 3  |   |   | ■ |   |   |  |   |
| 4  |   |   |   | ■ |   |  |   |
| 5  |   | ■ |   |   |   |  |   |
| 6  |   |   |   | ■ |   |  |   |
| 7  |   |   |   |   | ■ |  |   |
| 8  |   |   | ■ |   |   |  |   |
| 9  |   |   | ■ |   |   |  |   |
| 10 |   |   |   |   |   |  | ■ |
| 11 | ■ |   |   |   |   |  |   |
| 12 | ■ |   |   |   |   |  |   |
| 13 |   |   |   |   | ■ |  |   |
| 14 | ■ |   |   |   |   |  |   |
| 15 | ■ |   |   |   |   |  |   |
| 16 |   |   |   | ■ |   |  |   |

Risultato prova n. 14:

|    |  |  |  |  |  |  |    |
|----|--|--|--|--|--|--|----|
| 1  |  |  |  |  |  |  | 1  |
| 2  |  |  |  |  |  |  | 2  |
| 3  |  |  |  |  |  |  | 3  |
| 4  |  |  |  |  |  |  | 4  |
| 5  |  |  |  |  |  |  | 5  |
| 6  |  |  |  |  |  |  | 6  |
| 7  |  |  |  |  |  |  | 7  |
| 8  |  |  |  |  |  |  | 8  |
| 9  |  |  |  |  |  |  | 9  |
| 10 |  |  |  |  |  |  | 10 |
| 11 |  |  |  |  |  |  | 11 |
| 12 |  |  |  |  |  |  | 12 |
| 13 |  |  |  |  |  |  | 13 |
| 14 |  |  |  |  |  |  | 14 |
| 15 |  |  |  |  |  |  | 15 |
| 16 |  |  |  |  |  |  | 16 |

Risultato prova n. 15:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 16:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  |   |   | ■ |   |   |
| 2  | ■ |   |   |   |   |
| 3  |   | ■ |   |   |   |
| 4  | ■ |   |   |   |   |
| 5  |   |   |   | ■ |   |
| 6  |   |   | ■ |   |   |
| 7  | ■ |   |   |   |   |
| 8  | ■ |   |   |   |   |
| 9  |   |   |   | ■ |   |
| 10 |   |   |   |   | ■ |
| 11 |   |   | ■ |   |   |
| 12 | ■ |   |   |   |   |
| 13 |   |   |   | ■ |   |
| 14 |   |   | ■ |   |   |
| 15 | ■ |   |   |   |   |
| 16 |   | ■ |   |   |   |

Risultato prova n. 17:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 18:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 19:

|    |   |   |   |   |   |   |
|----|---|---|---|---|---|---|
| 1  |   |   |   |   |   |   |
| 2  |   |   |   |   |   |   |
| 3  |   |   | ■ | ■ |   |   |
| 4  | ■ | ■ |   |   |   |   |
| 5  | ■ | ■ |   |   |   |   |
| 6  |   |   | ■ | ■ |   |   |
| 7  |   |   |   |   | ■ | ■ |
| 8  | ■ | ■ |   |   |   |   |
| 9  |   |   |   | ■ | ■ |   |
| 10 | ■ | ■ |   |   |   |   |
| 11 |   |   |   | ■ | ■ |   |
| 12 |   |   | ■ | ■ |   |   |
| 13 | ■ | ■ |   |   |   |   |
| 14 | ■ | ■ |   |   |   |   |
| 15 |   |   |   | ■ | ■ |   |
| 16 |   |   |   |   | ■ |   |

Risultato prova n. 20:

## Risultato prova n. 21:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 22:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  | ■ | ■ |   |   |   |   |   |
| 3  | ■ | ■ |   |   |   |   |   |
| 4  |   |   |   |   | ■ | ■ |   |
| 5  | ■ | ■ |   |   |   |   |   |
| 6  |   |   |   | ■ | ■ |   |   |
| 7  | ■ | ■ |   |   |   |   |   |
| 8  |   |   |   |   |   | ■ | ■ |
| 9  |   |   |   |   |   | ■ | ■ |
| 10 |   |   |   |   | ■ | ■ |   |
| 11 |   |   |   | ■ | ■ |   |   |
| 12 |   |   | ■ | ■ |   |   |   |
| 13 | ■ | ■ |   |   |   |   |   |
| 14 |   |   | ■ | ■ |   |   |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 |   |   |   | ■ | ■ |   |   |

Risultato prova n. 23:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   |   |   |   | ■ |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  | ■ | ■ |   |   |   |   |   |
| 6  |   |   | ■ | ■ |   |   |   |
| 7  |   |   |   | ■ | ■ |   |   |
| 8  |   |   | ■ | ■ |   |   |   |
| 9  | ■ | ■ |   |   |   |   |   |
| 10 |   | ■ | ■ |   |   |   |   |
| 11 |   |   |   |   | ■ | ■ |   |
| 12 |   |   |   |   |   | ■ | ■ |
| 13 |   |   | ■ | ■ |   |   |   |
| 14 | ■ | ■ |   |   |   |   |   |
| 15 |   |   |   |   | ■ | ■ |   |
| 16 |   |   |   | ■ | ■ |   |   |

Risultato prova n. 24:

A 16x16 grid with red squares at various positions. The red squares are located at the following coordinates:

- (1, 1)
- (2, 2)
- (3, 3)
- (4, 4)
- (5, 5)
- (6, 6)
- (7, 7)
- (8, 8)
- (9, 9)
- (10, 10)
- (11, 11)
- (12, 12)
- (13, 13)
- (14, 14)
- (15, 15)
- (16, 16)
- (1, 15)
- (2, 14)
- (3, 13)
- (4, 12)
- (5, 11)
- (6, 10)
- (7, 9)
- (8, 8)
- (9, 7)
- (10, 6)
- (11, 5)
- (12, 4)
- (13, 3)
- (14, 2)
- (15, 1)
- (1, 1)

## Risultato prova n. 25:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 26:

|    |  |   |   |   |   |   |  |
|----|--|---|---|---|---|---|--|
| 1  |  | ■ |   |   |   |   |  |
| 2  |  |   | ■ |   |   |   |  |
| 3  |  | ■ |   |   |   |   |  |
| 4  |  |   |   | ■ |   |   |  |
| 5  |  |   | ■ |   |   |   |  |
| 6  |  | ■ |   |   |   |   |  |
| 7  |  |   | ■ |   |   |   |  |
| 8  |  |   |   | ■ |   |   |  |
| 9  |  |   |   |   | ■ |   |  |
| 10 |  | ■ |   |   |   |   |  |
| 11 |  |   |   | ■ |   |   |  |
| 12 |  |   |   |   |   | ■ |  |
| 13 |  |   |   |   |   | ■ |  |
| 14 |  |   |   |   | ■ |   |  |
| 15 |  | ■ |   |   |   |   |  |
| 16 |  | ■ |   |   |   |   |  |

Risultato prova n. 27:

A 16x16 grid with red squares at the following coordinates:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)

## Risultato prova n. 28:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 29:

|    |   |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|---|--|
| 1  |   |   | ■ |   |   |   |   |  |
| 2  | ■ | ■ |   |   |   |   |   |  |
| 3  | ■ | ■ |   |   |   |   |   |  |
| 4  | ■ | ■ |   |   |   |   |   |  |
| 5  |   |   |   |   | ■ | ■ |   |  |
| 6  | ■ | ■ |   |   |   |   |   |  |
| 7  | ■ | ■ |   |   |   |   |   |  |
| 8  |   |   | ■ | ■ |   |   |   |  |
| 9  |   |   |   | ■ | ■ |   |   |  |
| 10 |   |   | ■ | ■ |   |   |   |  |
| 11 | ■ | ■ |   |   |   |   |   |  |
| 12 |   |   |   |   |   | ■ | ■ |  |
| 13 |   |   |   |   | ■ | ■ |   |  |
| 14 |   |   |   | ■ | ■ |   |   |  |
| 15 |   |   |   |   |   | ■ | ■ |  |
| 16 |   |   |   | ■ | ■ |   |   |  |

Risultato prova n. 30:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 31:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 32:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 33:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  |   |   |   |   |   |
| 2  | ■ |   |   |   |   |
| 3  |   |   | ■ |   |   |
| 4  |   | ■ |   |   |   |
| 5  |   |   |   | ■ |   |
| 6  |   | ■ |   |   |   |
| 7  |   |   |   |   | ■ |
| 8  |   |   |   |   | ■ |
| 9  |   |   | ■ |   |   |
| 10 |   | ■ |   |   |   |
| 11 | ■ |   |   |   |   |
| 12 | ■ |   |   |   |   |
| 13 | ■ |   |   |   |   |
| 14 |   |   |   | ■ |   |
| 15 | ■ |   |   |   |   |
| 16 | ■ |   |   |   |   |

Risultato prova n. 34:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 35:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   | ■ | ■ |   |   |
| 3  |  | ■ |   |   |   |   |   |
| 4  |  |   | ■ |   |   |   |   |
| 5  |  |   |   | ■ | ■ |   |   |
| 6  |  |   |   |   |   | ■ | ■ |
| 7  |  |   |   |   | ■ | ■ |   |
| 8  |  | ■ |   |   |   |   |   |
| 9  |  |   |   | ■ | ■ |   |   |
| 10 |  | ■ |   |   |   |   |   |
| 11 |  |   |   | ■ | ■ |   |   |
| 12 |  | ■ | ■ |   |   |   |   |
| 13 |  | ■ | ■ |   |   |   |   |
| 14 |  |   |   |   | ■ | ■ |   |
| 15 |  |   |   |   |   | ■ | ■ |
| 16 |  |   |   |   |   | ■ | ■ |

Risultato prova n. 36:

|    |  |   |   |   |   |   |    |
|----|--|---|---|---|---|---|----|
| 1  |  |   |   |   |   |   | 1  |
| 2  |  | 1 |   |   |   |   | 2  |
| 3  |  |   |   |   |   |   | 3  |
| 4  |  | 1 |   |   |   |   | 4  |
| 5  |  |   |   |   |   | 1 | 5  |
| 6  |  | 1 |   |   |   |   | 6  |
| 7  |  | 1 |   |   |   |   | 7  |
| 8  |  |   | 1 | 1 |   |   | 8  |
| 9  |  |   | 1 | 1 |   |   | 9  |
| 10 |  | 1 |   |   |   |   | 10 |
| 11 |  |   |   |   | 1 |   | 11 |
| 12 |  |   | 1 | 1 |   |   | 12 |
| 13 |  |   |   |   |   | 1 | 13 |
| 14 |  | 1 |   |   |   |   | 14 |
| 15 |  |   |   | 1 | 1 |   | 15 |
| 16 |  |   |   | 1 | 1 |   | 16 |

Risultato prova n. 37:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   | ■ | ■ |   |   |   |
| 3  |  |   |   | ■ | ■ |   |   |
| 4  |  | ■ | ■ |   |   |   |   |
| 5  |  | ■ | ■ |   |   |   |   |
| 6  |  |   |   |   | ■ | ■ |   |
| 7  |  |   |   |   |   | ■ | ■ |
| 8  |  | ■ | ■ |   |   |   |   |
| 9  |  | ■ | ■ |   |   |   |   |
| 10 |  | ■ | ■ |   |   |   |   |
| 11 |  |   |   | ■ | ■ |   |   |
| 12 |  |   |   |   | ■ | ■ |   |
| 13 |  |   |   |   | ■ | ■ |   |
| 14 |  |   |   | ■ | ■ |   |   |
| 15 |  |   | ■ | ■ |   |   |   |
| 16 |  |   |   | ■ | ■ |   |   |

Risultato prova n. 38:

A 16x16 grid with red squares at the following coordinates:

- (3, 1)
- (2, 3)
- (4, 1)
- (5, 3)
- (6, 1)
- (7, 3)
- (8, 1)
- (9, 3)
- (10, 1)
- (11, 3)
- (12, 1)
- (13, 3)
- (14, 1)
- (15, 3)
- (16, 1)

## Risultato prova n. 39:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   | ■ | ■ |   |   |
| 3  |  |   |   | ■ | ■ |   |   |
| 4  |  |   |   |   | ■ |   |   |
| 5  |  | ■ | ■ |   |   |   |   |
| 6  |  | ■ | ■ |   |   |   |   |
| 7  |  |   |   |   |   | ■ | ■ |
| 8  |  |   | ■ | ■ |   |   |   |
| 9  |  |   |   |   | ■ | ■ |   |
| 10 |  | ■ | ■ |   |   |   |   |
| 11 |  |   |   |   | ■ | ■ |   |
| 12 |  |   |   |   |   | ■ | ■ |
| 13 |  | ■ | ■ |   |   |   |   |
| 14 |  | ■ | ■ |   |   |   |   |
| 15 |  |   | ■ | ■ |   |   |   |
| 16 |  |   | ■ | ■ |   |   |   |

Risultato prova n. 40:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 41:

|    |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|--|
| 1  |   | ■ |   |   |   |   |  |
| 2  |   |   |   | ■ |   |   |  |
| 3  |   |   | ■ |   |   |   |  |
| 4  |   |   |   |   |   | ■ |  |
| 5  |   |   |   |   | ■ |   |  |
| 6  |   |   |   |   | ■ |   |  |
| 7  |   |   |   |   |   | ■ |  |
| 8  |   |   |   | ■ |   |   |  |
| 9  | ■ |   |   |   |   |   |  |
| 10 | ■ |   |   |   |   |   |  |
| 11 | ■ |   |   |   |   |   |  |
| 12 | ■ |   |   |   |   |   |  |
| 13 |   |   |   | ■ |   |   |  |
| 14 | ■ |   |   |   |   |   |  |
| 15 |   |   | ■ |   |   |   |  |
| 16 |   |   | ■ |   |   |   |  |

Risultato prova n. 42:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  |   |   | ■ |   |   |
| 2  |   | ■ |   |   |   |
| 3  |   |   |   | ■ | ■ |
| 4  |   |   |   | ■ | ■ |
| 5  |   |   |   |   | ■ |
| 6  |   |   | ■ |   |   |
| 7  | ■ |   |   |   |   |
| 8  |   |   |   |   | ■ |
| 9  |   |   | ■ |   |   |
| 10 | ■ |   |   |   |   |
| 11 |   | ■ |   |   |   |
| 12 | ■ |   |   |   |   |
| 13 | ■ |   |   |   |   |
| 14 | ■ |   |   |   |   |
| 15 |   |   | ■ |   |   |
| 16 | ■ |   |   |   |   |

Risultato prova n. 43:

|    |   |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|---|--|
| 1  |   |   | ■ |   |   |   |   |  |
| 2  |   |   |   | ■ | ■ |   |   |  |
| 3  |   |   | ■ | ■ |   |   |   |  |
| 4  |   |   |   | ■ |   |   |   |  |
| 5  |   |   |   |   |   | ■ | ■ |  |
| 6  | ■ | ■ |   |   |   |   |   |  |
| 7  | ■ | ■ |   |   |   |   |   |  |
| 8  |   |   |   | ■ | ■ |   |   |  |
| 9  | ■ | ■ |   |   |   |   |   |  |
| 10 |   |   |   |   |   | ■ | ■ |  |
| 11 | ■ | ■ |   |   |   |   |   |  |
| 12 |   |   |   |   | ■ | ■ |   |  |
| 13 |   |   |   |   | ■ | ■ |   |  |
| 14 | ■ | ■ |   |   |   |   |   |  |
| 15 |   |   |   | ■ | ■ |   |   |  |
| 16 | ■ | ■ |   |   |   |   |   |  |

Risultato prova n. 44:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 45:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  |   |   |   |   |   |
| 2  | ■ |   |   |   |   |
| 3  |   | ■ |   |   |   |
| 4  |   |   | ■ |   |   |
| 5  |   |   |   | ■ |   |
| 6  |   |   |   |   | ■ |
| 7  |   |   |   |   | ■ |
| 8  |   |   |   |   | ■ |
| 9  | ■ |   |   |   |   |
| 10 | ■ |   |   |   |   |
| 11 | ■ |   |   |   |   |
| 12 |   |   | ■ |   |   |
| 13 |   |   | ■ |   |   |
| 14 | ■ |   |   |   |   |
| 15 | ■ |   |   |   |   |
| 16 |   |   |   |   | ■ |

Risultato prova n. 46:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 47:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  | ■ | ■ |   |   |   |   |   |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  | ■ | ■ |   |   |   |   |   |
| 6  |   |   |   | ■ | ■ |   |   |
| 7  | ■ | ■ |   |   |   |   |   |
| 8  |   |   |   |   | ■ | ■ |   |
| 9  |   |   |   |   |   | ■ | ■ |
| 10 |   |   |   |   |   | ■ | ■ |
| 11 | ■ | ■ |   |   |   |   |   |
| 12 | ■ | ■ |   |   |   |   |   |
| 13 |   |   | ■ | ■ |   |   |   |
| 14 |   |   | ■ | ■ |   |   |   |
| 15 |   |   |   | ■ | ■ |   |   |
| 16 |   |   |   |   | ■ | ■ |   |

Risultato prova n. 48:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  |   |   | ■ |   |   |
| 2  |   | ■ |   |   |   |
| 3  |   |   |   | ■ |   |
| 4  | ■ |   |   |   |   |
| 5  |   |   | ■ |   |   |
| 6  | ■ |   |   |   |   |
| 7  |   | ■ |   |   |   |
| 8  | ■ |   |   |   |   |
| 9  | ■ |   |   |   |   |
| 10 | ■ |   |   |   |   |
| 11 | ■ |   |   |   |   |
| 12 |   |   |   | ■ |   |
| 13 |   |   |   | ■ |   |
| 14 |   |   | ■ |   |   |
| 15 |   |   |   |   | ■ |
| 16 |   | ■ |   |   |   |

Risultato prova n. 49:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 50:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 51:

|    |   |  |   |   |   |   |    |
|----|---|--|---|---|---|---|----|
| 1  |   |  |   |   |   |   | 1  |
| 2  |   |  | 1 |   |   |   | 2  |
| 3  |   |  |   | 1 |   |   | 3  |
| 4  |   |  |   |   | 1 |   | 4  |
| 5  | 1 |  |   |   |   |   | 5  |
| 6  |   |  |   | 1 |   |   | 6  |
| 7  |   |  |   |   |   | 1 | 7  |
| 8  | 1 |  |   |   |   |   | 8  |
| 9  |   |  |   | 1 |   |   | 9  |
| 10 |   |  |   |   | 1 |   | 10 |
| 11 | 1 |  |   |   |   |   | 11 |
| 12 | 1 |  |   |   |   |   | 12 |
| 13 |   |  | 1 |   |   |   | 13 |
| 14 |   |  | 1 |   |   |   | 14 |
| 15 | 1 |  |   |   |   |   | 15 |
| 16 | 1 |  |   |   |   |   | 16 |

Risultato prova n. 52:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 53:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 54:

|    |  |   |   |   |   |   |  |
|----|--|---|---|---|---|---|--|
| 1  |  | ■ |   |   |   |   |  |
| 2  |  |   | ■ |   |   |   |  |
| 3  |  | ■ |   |   |   |   |  |
| 4  |  |   |   |   |   | ■ |  |
| 5  |  |   | ■ |   |   |   |  |
| 6  |  |   |   | ■ | ■ |   |  |
| 7  |  |   |   | ■ | ■ |   |  |
| 8  |  | ■ |   |   |   |   |  |
| 9  |  |   |   |   |   | ■ |  |
| 10 |  |   | ■ |   |   |   |  |
| 11 |  |   |   |   | ■ |   |  |
| 12 |  | ■ | ■ |   |   |   |  |
| 13 |  | ■ | ■ |   |   |   |  |
| 14 |  |   |   | ■ | ■ |   |  |
| 15 |  |   |   |   | ■ |   |  |
| 16 |  | ■ | ■ |   |   |   |  |

Risultato prova n. 55:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 56:

|    |   |  |   |   |   |   |  |
|----|---|--|---|---|---|---|--|
| 1  |   |  | ■ |   |   |   |  |
| 2  |   |  |   | ■ |   |   |  |
| 3  |   |  | ■ |   |   |   |  |
| 4  |   |  |   |   | ■ |   |  |
| 5  |   |  |   |   |   | ■ |  |
| 6  | ■ |  |   |   |   |   |  |
| 7  |   |  |   | ■ |   |   |  |
| 8  | ■ |  |   |   |   |   |  |
| 9  | ■ |  |   |   |   |   |  |
| 10 | ■ |  |   |   |   |   |  |
| 11 | ■ |  |   |   |   |   |  |
| 12 |   |  |   | ■ |   |   |  |
| 13 |   |  | ■ |   |   |   |  |
| 14 |   |  |   |   | ■ |   |  |
| 15 | ■ |  |   |   |   |   |  |
| 16 |   |  |   |   |   | ■ |  |

Risultato prova n. 57:

|    |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|--|
| 1  |   | ■ |   |   |   |   |  |
| 2  |   |   |   |   |   | ■ |  |
| 3  |   |   |   |   | ■ | ■ |  |
| 4  |   |   | ■ | ■ |   |   |  |
| 5  |   |   | ■ | ■ |   |   |  |
| 6  |   |   |   | ■ | ■ |   |  |
| 7  |   |   |   |   | ■ | ■ |  |
| 8  |   | ■ |   |   |   |   |  |
| 9  |   |   |   |   | ■ | ■ |  |
| 10 |   |   |   | ■ | ■ |   |  |
| 11 |   | ■ |   |   |   |   |  |
| 12 |   |   |   |   | ■ | ■ |  |
| 13 |   | ■ |   |   |   |   |  |
| 14 |   | ■ |   |   |   |   |  |
| 15 |   |   |   | ■ | ■ |   |  |
| 16 | ■ | ■ |   |   |   |   |  |

Risultato prova n. 58:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 59:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 60:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 61:

|    |   |   |   |   |
|----|---|---|---|---|
| 1  |   |   | ■ |   |
| 2  |   |   |   | ■ |
| 3  |   | ■ |   |   |
| 4  |   |   |   | ■ |
| 5  |   |   | ■ |   |
| 6  |   |   | ■ |   |
| 7  |   |   |   | ■ |
| 8  | ■ |   |   |   |
| 9  | ■ |   |   |   |
| 10 |   | ■ |   |   |
| 11 |   |   |   | ■ |
| 12 |   | ■ |   |   |
| 13 | ■ |   |   |   |
| 14 | ■ |   |   |   |
| 15 | ■ |   |   |   |
| 16 | ■ |   |   |   |

Risultato prova n. 62:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 63:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 64:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   | ■ | ■ |   |   |   |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  |   |   | ■ | ■ |   |   |   |
| 6  |   |   |   |   | ■ | ■ |   |
| 7  |   |   |   |   | ■ | ■ |   |
| 8  |   |   |   |   |   | ■ | ■ |
| 9  | ■ | ■ |   |   |   |   |   |
| 10 |   |   |   |   | ■ | ■ |   |
| 11 |   |   |   |   |   |   | ■ |
| 12 |   |   |   |   |   |   | ■ |
| 13 | ■ | ■ |   |   |   |   |   |
| 14 | ■ | ■ |   |   |   |   |   |
| 15 |   |   |   |   |   | ■ | ■ |
| 16 | ■ | ■ |   |   |   |   |   |

Risultato prova n. 65:

|    |   |   |   |   |   |   |    |
|----|---|---|---|---|---|---|----|
| 1  |   |   |   |   |   |   | 1  |
| 2  |   |   | 1 |   |   |   | 2  |
| 3  |   |   |   |   | 1 | 1 | 3  |
| 4  |   |   |   | 1 |   |   | 4  |
| 5  | 1 | 1 |   |   |   |   | 5  |
| 6  | 1 | 1 |   |   |   |   | 6  |
| 7  |   |   |   |   |   | 1 | 7  |
| 8  |   |   | 1 | 1 |   |   | 8  |
| 9  | 1 | 1 |   |   |   |   | 9  |
| 10 | 1 | 1 |   |   |   |   | 10 |
| 11 |   |   | 1 | 1 |   |   | 11 |
| 12 |   |   |   | 1 | 1 |   | 12 |
| 13 |   |   |   |   | 1 | 1 | 13 |
| 14 | 1 | 1 |   |   |   |   | 14 |
| 15 | 1 | 1 |   |   |   |   | 15 |
| 16 |   |   |   | 1 | 1 |   | 16 |

Risultato prova n. 66:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   | ■ | ■ |   |   |
| 3  |  | ■ |   |   |   |   |   |
| 4  |  |   | ■ |   |   |   |   |
| 5  |  |   |   |   |   | ■ | ■ |
| 6  |  | ■ |   |   |   |   |   |
| 7  |  |   |   | ■ | ■ |   |   |
| 8  |  |   |   |   | ■ | ■ |   |
| 9  |  |   |   | ■ | ■ |   |   |
| 10 |  | ■ |   |   |   |   |   |
| 11 |  |   |   |   | ■ | ■ |   |
| 12 |  | ■ | ■ |   |   |   |   |
| 13 |  | ■ | ■ |   |   |   |   |
| 14 |  |   | ■ | ■ |   |   |   |
| 15 |  |   | ■ | ■ |   |   |   |
| 16 |  |   |   |   |   | ■ |   |

Risultato prova n. 67:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 68:

|    |  |  |  |  |  |  |    |
|----|--|--|--|--|--|--|----|
| 1  |  |  |  |  |  |  | 1  |
| 2  |  |  |  |  |  |  | 2  |
| 3  |  |  |  |  |  |  | 3  |
| 4  |  |  |  |  |  |  | 4  |
| 5  |  |  |  |  |  |  | 5  |
| 6  |  |  |  |  |  |  | 6  |
| 7  |  |  |  |  |  |  | 7  |
| 8  |  |  |  |  |  |  | 8  |
| 9  |  |  |  |  |  |  | 9  |
| 10 |  |  |  |  |  |  | 10 |
| 11 |  |  |  |  |  |  | 11 |
| 12 |  |  |  |  |  |  | 12 |
| 13 |  |  |  |  |  |  | 13 |
| 14 |  |  |  |  |  |  | 14 |
| 15 |  |  |  |  |  |  | 15 |
| 16 |  |  |  |  |  |  | 16 |

Risultato prova n. 69:

|    |  |   |   |   |   |   |  |
|----|--|---|---|---|---|---|--|
| 1  |  | ■ |   |   |   |   |  |
| 2  |  |   |   | ■ |   |   |  |
| 3  |  | ■ |   |   |   |   |  |
| 4  |  | ■ |   |   |   |   |  |
| 5  |  |   |   | ■ |   |   |  |
| 6  |  |   |   |   |   | ■ |  |
| 7  |  |   | ■ |   |   |   |  |
| 8  |  | ■ |   |   |   |   |  |
| 9  |  | ■ |   |   |   |   |  |
| 10 |  |   | ■ |   |   |   |  |
| 11 |  |   |   |   | ■ |   |  |
| 12 |  |   |   |   |   | ■ |  |
| 13 |  |   | ■ |   |   |   |  |
| 14 |  |   |   | ■ |   |   |  |
| 15 |  | ■ |   |   |   |   |  |
| 16 |  |   |   |   | ■ |   |  |

Risultato prova n. 70:



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 1**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;

- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

5. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.  
D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.  
E. Nessuna delle precedenti.
- 

6. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

A. 101  

B. 000  

C. 110  

D. 010  

E. 100

```

---

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
    }
}

```

```

    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 25Exception in thread main MyExc1  
B. 2451  
C. 2Exception in thread main MyExc1  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

8. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC  
B. AAA  
C. CCC  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione

9. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente  
D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`  
E. Un array non possiede dei membri

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

A. 55
B. 5.55
C. 5.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.

```

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

12. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c)  
throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

13. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
  
    void remove() {  
        length--;  
        players[length] = null;  
    }  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

14. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

15. Qual'è l'output di questo codice?

```
String c = "I";  
void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    protected void f() {}  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B b) {  
        System.out.print(b.c);  
    }  
    public static void main(String[] args) {  
        B c = new B();  
        c.m((I) c);  
        c.m((A) c);  
        c.m(c);  
    }  
}
```

- A. IAC

- B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione**
  - E. Errore a tempo di esecuzione
- 

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
    finally {
```

```
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione**
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 2**

1. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

2. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

1. Tree g(String s) {
2. Park p = new Park();
3. Tree t = new Tree(s);
4. p.add(t);
5. t = null;
6. p.remove();
7. p = null;
8. return t;
9. }

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

3. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

4. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c)
throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}

- E. Nessuna delle precedenti.

---

5. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `sf new C.I()`.
- D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
- E. Nessuna delle precedenti.

---

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. 134

- B. 14

- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

---

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

9. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

A. IAC

B. AAA

C. CCC

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
```

```
String s = "Lemon";
public String type() {
    return s;
}
public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

12. Quale output si ottiene invocando il metodo m?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
    void q(Float f2, char [] a2, Float f5) {  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a1 == a2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f5 == f4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

---

13. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";
```

```
protected void m(I i) {  
    System.out.print(i.c);  
}  
protected void m(A a) {  
    System.out.print(a.c);  
}  
protected void m(B c) {  
    System.out.print(c.c);  
}  
public static void main(String[] args) {  
    I c = new B();  
    c.f();  
    ((A) c).f();  
    ((B) c).f();  
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {
```

```

        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 3**

1. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

2. Dato un file contenente il seguente codice:

```
package pk;

public class C {
```

```
static class I {
}
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

3. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `x = (Error) u;`
  - B. `g = (Exception) u;`
  - C. `x = (Error) g;`
  - D. `g = (Exception) x;`
  - E. Nessuno dei precedenti
- 

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                q();
                System.out.print(1);
            }
```

```

}
static void q() throws Exception {
    try {
        System.out.print(2);
        throw( new Exception() );
    }
    catch( MyExc2 r ) {
        System.out.print(3);
    }
    catch( MyExc1 c ) {
        System.out.print(4);
    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

#### 6. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

#### 7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
}

```

```

        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
 B. Lemon  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
 B. Lemon  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

9. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101  
 B. 000  
 C. 110  
 D. 010  
 E. 100
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
 C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente  
 D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`  
 E. Un array non possiede dei membri
- 

11. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A.  $u = c;$
  - B.  $c = (\text{Integer}) u;$
  - C.  $c = u;$
  - D.  $f = c;$
  - E.  $u = f;$
- 

12. Quale delle seguenti non è un sovraccaricamento corretto del metodo `m` nella classe `C`?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. static long m(long l){return 1;}
  - B. protected void m(int i, char c)  
throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 55
  - B. 55.5
  - C. 55.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class B1 {
```

```
    public static void main(String [] argv) {  
        try {  
            m();  
            System.out.print(1);  
        }  
        catch( MyExc1 t ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 i ) {  
        }  
        catch( Exception c ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc3() );  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(5);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 f ) {  
        }  
        catch( MyExc2 c ) {  
        }  
        catch( MyExc1 u ) {  
            System.out.print(6);  
        }  
    }  
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 55

- B. 5.5.5
  - C. 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

#### 16. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
  
    void remove() {  
        length--;  
        players[length] = null;  
    }  
}
```

}

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 4**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
```

```
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

---

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

#### 4. Qual è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

#### 5. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

#### 6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 7. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

---

8. Quale delle seguenti non è un sovraccaricamento corretto del metodo `m` nella classe `C`?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. `static long m(long l){return l;}`
- B. `protected void m(int i, char c)  
throws RuntimeException {}`
- C. `public final int m() {return 1;}`
- D. `public void m(int i){}`
- E. Nessuna delle precedenti.

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 55
- B. 5.5
- C. 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

10. Dato un file contenente il seguente codice:

```
package pk;  
  
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `C`.
- B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
- C. Allesterno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `sf new C.I()`.
- D. Un'istanza della classe `I` può essere costruita con `new C().new I()` anche fuori da `pk`.
- E. Nessuna delle precedenti.

---

11. Quale output si ottiene invocando il metodo `m`?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
    void q(Float f2, char [] a2, Float f5) {  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a1 == a2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f5 == f4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

---

12. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
}

```

```

static void p() throws Exception {
    try {
        throw( new MyExc1() );
    }
    catch( MyExc3 h ) {
        System.out.print(2);
    }
    catch( Exception h ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?
- 

```

public class A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
}

public static void main(String[] args) {
    long l = 11;
    char c = 2;
    System.out.print(l/c + " ");
    m(l, c);
}

```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 5**

1. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
  - B. `c = (Integer) u;`
  - C. `c = u;`
  - D. `f = c;`
  - E. `u = f;`
- 

2. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101

- B. 000
  - C. 110
  - D. 010
  - E. 100
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 55
- B. 5.5.5

- C. 5.5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

6. Qual è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
    }
}

```

```

}
catch( MyExc3 i ) {
}
catch( Exception c ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
    throw( new MyExc3() );
}
}

static void m() {
try {
    System.out.print(5);
    throw( new MyExc3() );
}
catch( MyExc3 f ) {
}
catch( MyExc2 c ) {
}
catch( MyExc1 u ) {
    System.out.print(6);
}
}
}

```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Dato un file contenente il seguente codice:

```

package pk;

public class C {
static class I {
}
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con sf new C.I().
  - D. Un'istanza della classe I può essere costruita con new C().new I() anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
  - E. Un array non possiede dei membri
- 

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

12. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c)
throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

13. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

14. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {

```

```

        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

A. Tree
B. Lemon
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

16. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }
}

```

```

void remove() {
    length--;
    players[length] = null;
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 6**

1. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c) throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

2. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

3. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
```

```
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
```

```

}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 5. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
- 

#### 6. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        trees[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

#### 7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 55
  - B. 55.5
  - C. 55.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
}

```

```

}

static void q() throws Exception {
    try {
        System.out.print(2);
        throw( new Exception() );
    }
    catch( MyExc2 r ) {
        System.out.print(3);
    }
    catch( MyExc1 c ) {
        System.out.print(4);
    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 25Exception in thread main MyExc1  
B. 2451  
C. 2Exception in thread main MyExc1  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente  
D. Non tutti i tipi di eccezioni estendono la classe RuntimeException  
E. Un array non possiede dei membri
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {

```

```

        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
A. Tree
B. Lemon
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.
```

---

11. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101  
B. 000  
C. 110  
D. 010  
E. 100
- 

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {

```

```

        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

---

15. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

A. 5 5
B. 5 5.5
C. 5 5.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.
```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 7**

1. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

2. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

A. static long m(long l){return l;}
B. protected void m(int i, char c)
throws RuntimeException {}
C. public final int m() {return 1;}
D. public void m(int i){}
E. Nessuna delle precedenti.
```

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14

- C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

4. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {

```

```
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
}

```

```

Tree() {
    type();
}
private void type() {
    System.out.println(s);
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

11. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`

E. u = f;

---

12. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

13. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
```

```
protected void m(I i) {
    System.out.print(i.c);
}
protected void m(A a) {
    System.out.print(a.c);
}
protected void m(B b) {
    System.out.print(b.c);
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
}

public static void main(String[] args) {
    long l = 11;
    int i = 2;
    System.out.print(l/i + " ");
    m(l, i);
}

A. 5.5
B. 5.55
C. 5.55d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.
```

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
```

```

        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

A. 5 5
B. 5 5.5
C. 5 5.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.

```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 8**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

4. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }

    A. static long m(long l){return l;}
    B. protected void m(int i, char c)
        throws RuntimeException {}
```

- C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

### 5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

### 6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
}

```

```

private void type() {
    System.out.println(s);
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

A. Tree
B. Lemon
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

### 7. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}

class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.

- D. Al termine della linea 8.  
E. Non si può dire, poichè loggetto viene restituito dal metodo g.
- 

8. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente  
D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`  
E. Un array non possiede dei membri
- 

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514  
B. 514Exception in thread main MyExc3  
C. 51  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC  
B. AAA  
C. CCC  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione
- 

11. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `sf new C.I()`.
- D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
- E. Nessuna delle precedenti.

---

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

13. Quale output si ottiene invocando il metodo m?

```
class B {  
  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
    void q(Float f2, char [] a2, Float f5) {  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a1 == a2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(f5 == f4) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
}  
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

---

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

15. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- }
- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

16. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 9**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
}

public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;

- B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

3. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

4. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

5. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

6. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
  - B. `c = (Integer) u;`
  - C. `c = u;`
  - D. `f = c;`
  - E. `u = f;`
- 

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

8. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. `static long m(long l){return l;}`
  - B. `protected void m(int i, char c)`  
`throws RuntimeException {}`
  - C. `public final int m() {return 1;}`
  - D. `public void m(int i){}`
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 55  
 B. 55.5  
 C. 55.5d  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 55  
 B. 55.5  
 C. 55.5d  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti.
- 

11. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {

```

```

        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC  
 B. AAA  
 C. CCC  
 D. Errore a tempo di compilazione  
 E. Errore a tempo di esecuzione
- 

12. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

1. Tree g(String s) {
2. Park p = new Park();
3. Tree t = new Tree(s);

```

4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

13. Qual è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
    finally {

```

```
        System.out.print(5);
    }
}
```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
```

```
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 10

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

2. Dato un file contenente il seguente codice:

```
package pk;  
  
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
- D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
- E. Nessuna delle precedenti.

3. Quale output si ottiene invocando il metodo m?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
    void q(Float f2, char [] a2, Float f5) {  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a1 == a2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f5 == f4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv) {
```

```

try {
    m();
    System.out.print(1);
}
catch( MyExc1 t ) {
    System.out.print(2);
}
catch( MyExc3 i ) {
}
catch( Exception c ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
    throw( new MyExc3() );
}
}

static void m() {
try {
    System.out.print(5);
    throw( new MyExc3() );
}
catch( MyExc3 f ) {
}
catch( MyExc2 c ) {
}
catch( MyExc1 u ) {
    System.out.print(6);
}
}
}

```

- A. 514  
B. 514Exception in thread main MyExc3  
C. 51  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
}

```

```

    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
A. Tree
B. Lemon
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

6. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
A. static long m(long l){return l;}
B. protected void m(int i, char c)
    throws RuntimeException {}
C. public final int m() {return 1;}
D. public void m(int i){}
E. Nessuna delle precedenti.

```

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
}

```

```

public static void main(String[] args) {
    long l = 11;
    char c = 2;
    System.out.print(l/c + " ");
    m(l, c);
}

```

- A. 5 5  
B. 5 5.5  
C. 5 5.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

8. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

9. Qual è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {

```

```

String c = "B";
protected void m(I i) {
    System.out.print(i.c);
}
protected void m(A a) {
    System.out.print(a.c);
}
protected void m(B c) {
    System.out.print(c.c);
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

11. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
- 

12. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

13. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                q();
                System.out.print(1);
            }
        }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
}

```

```
public static void main(String[] args) {  
    Plant a = new Lemon();  
    a.type();  
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class C1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( MyExc2 t ) {  
            throw( new MyExc1() );  
        }  
    }  
}
```

```
static void p() throws Exception {  
    try {  
        throw( new MyExc1() );  
    }  
    catch( MyExc3 h ) {  
        System.out.print(2);  
    }  
    catch( Exception h ) {  
        System.out.print(3);  
    }  
    finally {  
        System.out.print(4);  
    }  
}
```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 11**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
}

public static void main(String[] args) {
    long l = 11;
    int i = 2;
    System.out.print(l/i + " ");
    m(l, i);
}
}

A. 55
B. 55.5
C. 55.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.
```

2. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}
```

```
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 25Exception in thread main MyExc1  
 B. 2451  
 C. 2Exception in thread main MyExc1  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514  
 B. 514Exception in thread main MyExc3  
 C. 51  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

```
public class A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
}
public static void main(String[] args) {
    long l = 11;
    char c = 2;
    System.out.print(l/c + " ");
    m(l, c);
}
A. 55
B. 55.5
C. 55.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.
```

---

6. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.  
 B. Al termine della linea 6.  
 C. Al termine della linea 7.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

- D. Al termine della linea 8.  
E. Non si può dire, poichè loggetto viene restituito dal metodo g.
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente  
D. Non tutti i tipi di eccezioni estendono la classe RuntimeException  
E. Un array non possiede dei membri
- 

8. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B c) {  
        System.out.print(c.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();  
        c.f();  
        ((A) c).f();  
        ((B) c).f();  
    }  
}
```

- A. IAC  
B. AAA

- C. CCC  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione
- 

9. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. static long m(long l){return l;}  
B. protected void m(int i, char c)  
throws RuntimeException {}  
C. public final int m() {return 1;}  
D. public void m(int i){}  
E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

11. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
    }
}

```

- a.type();  
}
  - A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `x = (Error) u;`
  - B. `g = (Exception) u;`
  - C. `x = (Error) g;`
  - D. `g = (Exception) x;`
  - E. Nessuno dei precedenti
- 

15. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- 

C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `sf new C.I()`.

D. Un'istanza della classe I può essere costruita con `C().new I()` anche fuori da pk.

E. Nessuna delle precedenti.

---

16. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
  - B. `c = (Integer) u;`
  - C. `c = u;`
  - D. `f = c;`
  - E. `u = f;`
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 12**

1. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree

- B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

4. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

5. Qual'è l'output di questo codice?

```
String c = "I";  
void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    protected void f() {}  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B b) {  
        System.out.print(b.c);  
    }  
    public static void main(String[] args) {  
        B c = new B();  
        c.m((I) c);  
        c.m((A) c);  
        c.m(c);  
    }  
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}  
  
A. 5 5  
B. 5 5.5  
C. 5 5.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
```

7. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

8. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}  
  
A. static long m(long l){return l;}  
B. protected void m(int i, char c)  
    throws RuntimeException {}  
C. public final int m() {return 1;}  
D. public void m(int i){}
```

E. Nessuna delle precedenti.

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            m();  
            System.out.print(1);  
        }  
        catch( MyExc1 t ) {  
            System.out.print(2);  
        }  
    }  
}
```

```
}  
catch( MyExc3 i ) {  
}  
catch( Exception c ) {  
    System.out.print(3);  
}  
finally {  
    System.out.print(4);  
    throw( new MyExc3() );  
}  
}  
static void m() {  
    try {  
        System.out.print(5);  
        throw( new MyExc3() );  
    }  
    catch( MyExc3 f ) {  
    }  
    catch( MyExc2 c ) {  
    }  
    catch( MyExc1 u ) {  
        System.out.print(6);  
    }  
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
        }
    }
}

```

```

        p();
    }
    catch( MyExc2 t ) {
        throw( new MyExc1() );
    }
}
static void p() throws Exception {
    try {
        throw( new MyExc1() );
    }
    catch( MyExc3 h ) {
        System.out.print(2);
    }
    catch( Exception h ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

16. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";
```

```
protected void m(I i) {  
    System.out.print(i.c);  
}  
protected void m(A a) {  
    System.out.print(a.c);  
}  
protected void m(B c) {  
    System.out.print(c.c);  
}  
public static void main(String[] args) {  
    I c = new B();  
    c.f();  
    ((A) c).f();  
    ((B) c).f();  
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 13**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

3. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `x = (Error) u;`
  - B. `g = (Exception) u;`
  - C. `x = (Error) g;`
  - D. `g = (Exception) x;`
  - E. Nessuno dei precedenti
- 

4. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`

E. u = f;

- 
5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

6. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
  
    void remove() {  
        length--;  
        players[length] = null;  
    }  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.  
B. Al termine della linea 6.  
C. Al termine della linea 7.  
D. Al termine della linea 8.  
E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 55  
B. 5.5  
C. 5.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo codice?

```
String c = "I";  
void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    protected void f() {}  
    abstract void m(A a);  
}
```

```

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

#### 9. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101

- B. 000
  - C. 110
  - D. 010
  - E. 100
- 

#### 10. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

#### 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
}

```

```

    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

**12. Qual'è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione

D. 14Exception in thread main MyExc1

E. Nessuna delle precedenti

---

**13. Dire quale delle seguenti affermazioni è vera:**

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
  - E. Un array non possiede dei membri
- 

**14. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?**

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c)
 throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

**15. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 55
- B. 5.5
- C. 5.5d

D. Errore a tempo di compilazione

E. Nessuna delle precedenti.

---

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

```
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}
```

A. 514

B. 514Exception in thread main MyExc3

C. 51

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 14**

1. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
```

```

Tree() {
    type();
}
void type() {
    System.out.println(s);
}
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 55  
B. 55.5  
C. 55.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

```

```

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

7. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

1. Tree g(String s) {
2. Park p = new Park();
3. Tree t = new Tree(s);
4. p.add(t);
5. t = null;
6. p.remove();
7. p = null;
8. return t;
9. }

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

8. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
- 

9. Qual'è l'output di questo codice?

```
String c = "I";  
void f();  
  
}  
  
abstract class A implements I {  
    String c = "A";  
    protected void f() {}  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B b) {  
        System.out.print(b.c);  
    }  
    public static void main(String[] args) {  
        B c = new B();  
        c.m((I) c);  
        c.m((A) c);  
        c.m(c);  
    }  
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
  - E. Un array non possiede dei membri
- 

11. Dato un file contenente il seguente codice:

```
package pk;
```

```
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class C1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( MyExc2 t ) {
```

```

        throw( new MyExc1() );
    }
}

static void p() throws Exception {
    try {
        throw( new MyExc1() );
    }
    catch( MyExc3 h ) {
        System.out.print(2);
    }
    catch( Exception h ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 134  
 B. 14  
 C. Errore a tempo di compilazione  
 D. 14Exception in thread main MyExc1  
 E. Nessuna delle precedenti
- 

### 13. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101  
 B. 000

C. 110

D. 010

E. 100

---

### 14. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;  
 B. g = (Exception) u;  
 C. x = (Error) g;  
 D. g = (Exception) x;  
 E. Nessuno dei precedenti
- 

### 15. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
    }
}
```

```
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 15**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Dato un file contenente il seguente codice:

```
package pk;
```

```
public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

3. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `x = (Error) u;`
  - B. `g = (Exception) u;`
  - C. `x = (Error) g;`
  - D. `g = (Exception) x;`
  - E. Nessuno dei precedenti
- 

4. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}
```

```
abstract class A implements I {
    String c = "A";
```

```

protected void f() {}
abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

5. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return 1;}
  - B. protected void m(int i, char c) throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);

```

```

                p();
            }
            catch( MyExc2 t ) {
                throw( new MyExc1() );
            }
        }
        static void p() throws Exception {
            try {
                throw( new MyExc1() );
            }
            catch( MyExc3 h ) {
                System.out.print(2);
            }
            catch( Exception h ) {
                System.out.print(3);
            }
            finally {
                System.out.print(4);
            }
        }
    }

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                q();
                System.out.print(1);
            }
            static void q() throws Exception {
                try {
                    System.out.print(2);
                    throw( new Exception() );
                }
                catch( MyExc2 r ) {
                    System.out.print(3);
                }
                catch( MyExc1 c ) {
                    System.out.print(4);
                }
                catch( Exception g ) {
                    throw( new MyExc1() );
                }
                finally {
                    System.out.print(5);
                }
            }

```

```
    }
}
```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
```

```
Tree() {
    type();
}
private void type() {
    System.out.println(s);
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

13. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

14. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
  
    void remove() {  
        length--;  
        players[length] = null;  
    }  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

15. Qual è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}
```

```
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}
```

```
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B c) {  
        System.out.print(c.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();  
        c.f();  
        ((A) c).f();  
        ((B) c).f();  
    }  
}
```

```
}  
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

16. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 16**

1. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

2. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c) throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

4. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

---

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

6. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
    }
}
```

```
    players[length] = null;
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
```

```
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
- E. Un array non possiede dei membri

11. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
```

```

        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

A. 55
B. 5.5
C. 5.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.

```

---

14. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
- 

15. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {

```

```

        System.out.print(i.c);
    }
protected void m(A a) {
    System.out.print(a.c);
}
protected void m(B c) {
    System.out.print(c.c);
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 17**

1. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

A. 5.5
B. 5.5.5
C. 5.5.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.
```

3. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        }
    }
}
```

```

    } else {
        System.out.print(0);
    }
}

```

- A. 101  
 B. 000  
 C. 110  
 D. 010  
 E. 100
- 

#### 4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1  
 B. 2451  
 C. 2Exception in thread main MyExc1  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC  
 B. AAA  
 C. CCC  
 D. Errore a tempo di compilazione  
 E. Errore a tempo di esecuzione
- 

#### 6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
    }
}

```

#### 5. Qual'è l'output di questo codice?

```

        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}

```

- A. 514  
B. 514Exception in thread main MyExc3  
C. 51  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

7. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;  
B. g = (Exception) u;  
C. x = (Error) g;  
D. g = (Exception) x;  
E. Nessuno dei precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
}

```

```

    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134  
B. 14  
C. Errore a tempo di compilazione  
D. 14Exception in thread main MyExc1  
E. Nessuna delle precedenti

---

10. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `sf new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

12. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

A. 5.5
B. 5.5.5
C. 5.5.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.
```

---

14. Qual è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
```

```

        System.out.print(a.c);
    }
protected void m(B b) {
    System.out.print(b.c);
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

15. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

---

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
- 

16. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return 1;}
  - B. protected void m(int i, char c) throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 18**

1. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

2. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
}
```

```
public void f() {
    System.out.print(c);
}
abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

- 
4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5  
B. 5 5.5  
C. 5 5.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                q();  
                System.out.print(1);  
            }  
            static void q() throws Exception {  
                try {  
                    System.out.print(2);  
                    throw( new Exception() );  
                }  
                catch( MyExc2 r ) {  
                    System.out.print(3);  
                }  
                catch( MyExc1 c ) {  
                    System.out.print(4);  
                }  
                catch( Exception g ) {  
                    throw( new MyExc1() );  
                }  
                finally {  
                    System.out.print(5);  
                }  
            }  
        }
```

- A. 25Exception in thread main MyExc1  
B. 2451  
C. 2Exception in thread main MyExc1  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

6. Dato un file contenente il seguente codice:

```
package pk;  
  
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.  
D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.  
E. Nessuna delle precedenti.
- 

7. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
  
    void remove() {  
        length--;  
        players[length] = null;  
    }  
}
```

e il metodo g

1. Tree g(String s) {
2. Park p = new Park();
3. Tree t = new Tree(s);
4. p.add(t);
5. t = null;
6. p.remove();
7. p = null;
8. return t;
9. }

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

8. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

9. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
- 

10. Qual è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
```

```
}
```

```
class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}
```

---

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
    finally {
        System.out.print(4);
    }
}
```

---

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    void type() {  
        System.out.println(s);  
    }  
}
```

```
class Lemon extends Tree implements Plant{  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}  
  
A. 5.5  
B. 5.5.5  
C. 5.5.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
```

15. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

---

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        } catch( MyExc1 t ) {
            System.out.print(2);
        } catch( MyExc3 i ) {
        } catch( Exception c ) {
            System.out.print(3);
        } finally {
            System.out.print(4);
            throw( new MyExc3() );
    }
}
```

```
}
```

```
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    } catch( MyExc3 f ) {
    } catch( MyExc2 c ) {
    } catch( MyExc1 u ) {
        System.out.print(6);
    }
}
```

- 
- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 19**

1. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
}
```

```
public static void main(String[] args) {
    long l = 11;
    char c = 2;
    System.out.print(l/c + " ");
    m(l, c);
}
```

- A. 55
- B. 5.55
- C. 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
}
public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

4. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

5. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        trees[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 55
  - B. 55.5
  - C. 55.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            m();  
            System.out.print(1);  
        }  
        catch( MyExc1 t ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 i ) {  
        }  
        catch( Exception c ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc3() );  
        }  
    }  
  
    static void m() {  
        try {  
            System.out.print(5);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 f ) {  
        }  
        catch( MyExc2 c ) {  
        }  
    }  
}
```

```
}  
catch( MyExc1 u ) {  
    System.out.print(6);  
}  
}  
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo codice?

```
String c = "I";  
void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    protected void f() {}  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B b) {  
        System.out.print(b.c);  
    }  
    public static void main(String[] args) {  
        B c = new B();  
        c.m((I) c);  
        c.m((A) c);  
        c.m(c);  
    }  
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

11. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

**12. Qual'è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

**13. Date le dichiarazioni:**

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
- B. `c = (Integer) u;`

- C. `c = u;`
  - D. `f = c;`
  - E. `u = f;`
- 

**14. Qual'è l'output di questo codice?**

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

**15. Date le dichiarazioni:**

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `x = (Error) u;`

- B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
```

```
        }
    }
    catch( MyExc2 r ) {
        System.out.print(3);
    }
    catch( MyExc1 c ) {
        System.out.print(4);
    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 20

1. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class C1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( MyExc2 t ) {  
            throw( new MyExc1() );  
        }  
    }  
    static void p() throws Exception {  
        try {  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 h ) {  
            System.out.print(2);  
        }  
        catch( Exception h ) {  
            System.out.print(3);  
        }  
    finally {  
        System.out.print(4);  
    }  
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

3. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

4. Qual'è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }
```

```

public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

## 5. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

## 6. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
  - E. Un array non possiede dei membri
- 

## 7. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

## 8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

A. 5 5
B. 5 5.5
C. 5 5.5d

```

- D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

```
        ((A) c).f();  
        ((B) c).f();  
    }  
}
```

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 55  
B. 55.5  
C. 55.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

10. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B c) {  
        System.out.print(c.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();  
        c.f();  
    }  
}
```

- A. IAC  
B. AAA  
C. CCC  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione
- 

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            q();  
            System.out.print(1);  
        }  
        static void q() throws Exception {  
            try {  
                System.out.print(2);  
                throw( new Exception() );  
            }  
            catch( MyExc2 r ) {  
                System.out.print(3);  
            }  
            catch( MyExc1 c ) {  
                System.out.print(4);  
            }  
            catch( Exception g ) {  
                throw( new MyExc1() );  
            }  
            finally {  
                System.out.print(5);  
            }  
        }  
    }
```

- A. 25Exception in thread main MyExc1  
B. 2451  
C. 2Exception in thread main MyExc1  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**13. Qual'è l'output di questo codice?**

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {

```

```

        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

**14. Quale output si ottiene invocando il metodo m?**

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

---

15. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
```

```
}
```

```
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
```

```
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 21**

1. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

2. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
```

```
a3 = new char [5];
q(f3, a3, f4);
}
void q(Float f2, char [] a2, Float f5) {
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a1 == a2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f5 == f4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
}

public static void main(String[] args) {
    long l = 11;
    char c = 2;
    System.out.print(l/c + " ");
    m(l, c);
}
```

- A. 55
- B. 5 5.5

- C. 5.5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

#### 4. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
  
    void remove() {  
        length--;  
        players[length] = null;  
    }  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

#### 5. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c) throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

#### 6. Dato un file contenente il seguente codice:

```
package pk;  
  
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

#### 7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                q();  
                System.out.print(1);  
            }  
            static void q() throws Exception {  
                try {  
                    System.out.print(2);  
                    throw( new Exception() );  
                }  
                catch( MyExc2 r ) {  
                    System.out.print(3);  
                }  
                catch( MyExc1 c ) {  
                    System.out.print(4);  
                }  
                catch( Exception g ) {  
                    throw( new MyExc1() );  
                }  
            }  
        }  
}
```

```

        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1  
B. 2451  
C. 2Exception in thread main MyExc1  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134  
B. 14  
C. Errore a tempo di compilazione  
D. 14Exception in thread main MyExc1  
E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;  
B. c = (Integer) u;  
C. c = u;  
D. f = c;  
E. u = f;
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

11. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

```

```

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

**12. Qual'è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {

```

```

        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

A. 514
B. 514Exception in thread main MyExc3
C. 51
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
}
public static void main(String[] args) {

```

```
long l = 11;
int i = 2;
System.out.print(l/i + " ");
m(l, i);
}
```

- A. 5 5  
B. 5 5.5  
C. 5 5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

15. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;  
B. g = (Exception) u;  
C. x = (Error) g;  
D. g = (Exception) x;  
E. Nessuno dei precedenti
- 

16. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente  
D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`  
E. Un array non possiede dei membri
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 22**

1. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

2. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

3. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}

- B. `protected void m(int i, char c)`  
throws `RuntimeException {}`  
C. `public final int m() {return 1;}`  
D. `public void m(int i){}`  
E. Nessuna delle precedenti.

---

4. Date le dichiarazioni:

```
Exception g;  
Object u;  
Error x;  
g = new Exception();  
x = new Error();  
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `x = (Error) u;`  
B. `g = (Exception) u;`  
C. `x = (Error) g;`  
D. `g = (Exception) x;`  
E. Nessuno dei precedenti

---

5. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B c) {  
        System.out.print(c.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();  
        c.f();  
        ((A) c).f();  
        ((B) c).f();  
    }  
}
```

- A. IAC  
B. AAA  
C. CCC  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione

---

6. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente  
D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`  
E. Un array non possiede dei membri

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 55  
B. 5.5  
C. 5.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.

---

8. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A.  $u = c;$
  - B.  $c = (\text{Integer}) u;$
  - C.  $c = u;$
  - D.  $f = c;$
  - E.  $u = f;$
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}  
  
A. 55  
B. 5 5.5  
C. 5 5.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
```

---

10. Qual'è l'output di questo codice?

```
String c = "I";  
void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    protected void f() {}  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B b) {  
        System.out.print(b.c);  
    }  
}
```

---

```
}  
public static void main(String[] args) {  
    B c = new B();  
    c.m((I) c);  
    c.m((A) c);  
    c.m(c);  
}
```

---

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            q();  
            System.out.print(1);  
        }  
        static void q() throws Exception {  
            try {  
                System.out.print(2);  
                throw( new Exception() );  
            }  
            catch( MyExc2 r ) {  
                System.out.print(3);  
            }  
            catch( MyExc1 c ) {  
                System.out.print(4);  
            }  
            catch( Exception g ) {  
                throw( new MyExc1() );  
            }  
            finally {  
                System.out.print(5);  
            }  
        }  
    }
```

---

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**13. Dato un file contenente il seguente codice:**

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

**15. Qual'è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {

```

```

        throw( new MyExc1() );
    }
    catch( MyExc3 h ) {
        System.out.print(2);
    }
    catch( Exception h ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 134  
B. 14  
C. Errore a tempo di compilazione  
D. 14Exception in thread main MyExc1  
E. Nessuna delle precedenti
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

```

```

}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 23**

1. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

2. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
```

```
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010

E. 100

4. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

5. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;

E. Nessuno dei precedenti

6. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

A. static long m(long l){return l;}
B. protected void m(int i, char c)
throws RuntimeException {}
C. public final int m() {return 1;}
D. public void m(int i){}
E. Nessuna delle precedenti.
```

7. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
- D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
- E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }
}

public static void main(String[] args) {
    long l = 11;
    char c = 2;
    System.out.print(l/c + " ");
    m(l, c);
}
```

- A. 55
  - B. 55.5
  - C. 55.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
    }
}

```

```

        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}

```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
}

public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}

```

```

    }
}

A. Tree
B. Lemon
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

```

```

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

15. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;
    void add(Tree t) {
        trees[length++] = t;
    }
}

```

```
}

void remove() {
    length--;
    players[length] = null;
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

A. 5 5
B. 5 5.5
C. 5 5.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.
```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 24**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 55
  - B. 55.5
  - C. 55.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
```

```

    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 5. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

#### 6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

#### 7. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

---

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

8. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

9. Date le dichiarazioni:

```
Exception g;  
Object u;  
Error x;  
g = new Exception();  
x = new Error();  
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `x = (Error) u;`
  - B. `g = (Exception) u;`
  - C. `x = (Error) g;`
  - D. `g = (Exception) x;`
  - E. Nessuno dei precedenti
- 

10. Dato un file contenente il seguente codice:

```
package pk;  
  
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 55
  - B. 55.5
  - C. 55.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

13. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c)  
throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            m();  
            System.out.print(1);  
        }  
        catch( MyExc1 t ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 i ) {  
            System.out.print(3);  
        }  
    }  
}
```

```
}  
catch( Exception c ) {  
    System.out.print(3);  
}  
finally {  
    System.out.print(4);  
    throw( new MyExc3() );  
}  
}  
static void m() {  
    try {  
        System.out.print(5);  
        throw( new MyExc3() );  
    }  
    catch( MyExc3 f ) {  
    }  
    catch( MyExc2 c ) {  
    }  
    catch( MyExc1 u ) {  
        System.out.print(6);  
    }  
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B c) {  
        System.out.print(c.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();  
    }  
}
```

```
c.f();  
((A) c).f();  
((B) c).f();  
}  
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

16. Quale output si ottiene invocando il metodo m?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
}
```

```
void q(Float f2, char [] a2, Float f5) {  
    if(f2 == f1) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(a1 == a2) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(f5 == f4) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
}
```

- 
- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 25**

1. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

2. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
```

```
a3 = new char [5];
q(f3, a3, f4);
}
void q(Float f2, char [] a2, Float f5) {
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(a1 == a2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f5 == f4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

4. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

5. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c)
throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

6. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        trees[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

8. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `sf new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
}

public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
- B. `c = (Integer) u;`
- C. `c = u;`
- D. `f = c;`
- E. `u = f;`

- 
12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 55  
B. 55.5  
C. 55.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

13. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B c) {  
        System.out.print(c.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();  
        c.f();  
        ((A) c).f();  
        ((B) c).f();  
    }  
}
```

}

- A. IAC  
B. AAA  
C. CCC  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
    }  
}
```

```
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {

```

```
                throw( new MyExc1() );
            }
        }
        static void p() throws Exception {
            try {
                throw( new MyExc1() );
            }
            catch( MyExc3 h ) {
                System.out.print(2);
            }
            catch( Exception h ) {
                System.out.print(3);
            }
            finally {
                System.out.print(4);
            }
        }
    }
}
```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 26**

1. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

2. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
```

```
abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
    }
}
```

```
    players[length] = null;
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

---

4. Dato un file contenente il seguente codice:

```
package pk;

public class C {
static class I {
}
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
- D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
- E. Nessuna delle precedenti.

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {

```

```
                q();
                System.out.print(1);
            }
        }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {
```

```
    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args){
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5  
 B. 5 5.5  
 C. 5 5.5d  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti.
- 

8. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}  
 B. protected void m(int i, char c)  
 throws RuntimeException {}  
 C. public final int m() {return 1;}  
 D. public void m(int i){}  
 E. Nessuna delle precedenti.
- 

9. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {

```

```

        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC  
 B. AAA  
 C. CCC  
 D. Errore a tempo di compilazione  
 E. Errore a tempo di esecuzione
- 

10. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;  
 B. g = (Exception) u;  
 C. x = (Error) g;  
 D. g = (Exception) x;  
 E. Nessuno dei precedenti
- 

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {

```

```

        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514  
B. 514Exception in thread main MyExc3  
C. 51  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
B. Lemon

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

13. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente  
D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`  
E. Un array non possiede dei membri
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- }
- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 27**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
        }
    }
}
```

```

        throw( new Exception() );
    }
    catch( MyExc2 r ) {
        System.out.print(3);
    }
    catch( MyExc1 c ) {
        System.out.print(4);
    }
    catch( Exception g ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 25Exception in thread main MyExc1  
B. 2451  
C. 2Exception in thread main MyExc1  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

4. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente  
D. Non tutti i tipi di eccezioni estendono la classe RuntimeException  
E. Un array non possiede dei membri
- 

5. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}  
B. protected void m(int i, char c)  
throws RuntimeException {}  
C. public final int m() {return 1;}  
D. public void m(int i){}  
E. Nessuna delle precedenti.
- 

6. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;  
B. g = (Exception) u;  
C. x = (Error) g;  
D. g = (Exception) x;  
E. Nessuno dei precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5  
B. 5 5.5  
C. 5 5.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

8. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;
}

```

```

void add(Tree t) {
    trees[length++] = t;
}

void remove() {
    length--;
    players[length] = null;
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}

public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
    }
}

```

```

        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

A. 55
B. 55.5
C. 55.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.

```

---

12. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

14. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

15. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `sf new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}
```

```
class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 28**

1. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

2. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

3. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}
```

```
abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}
```

```
class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
```

```

        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
    }
}

```

```

    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514  
B. 514Exception in thread main MyExc3  
C. 51  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;  
B. g = (Exception) u;  
C. x = (Error) g;  
D. g = (Exception) x;  
E. Nessuno dei precedenti
- 

9. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        }
    }
}

```

```

    } else {
        System.out.print(0);
    }
    if(f5 == f4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 101  
B. 000  
C. 110  
D. 010  
E. 100
- 

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1  
B. 2451  
C. 2Exception in thread main MyExc1  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

12. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
- B. AAA

C. CCC

- D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

15. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `sf new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
}
```

```
    }
}
static void p() throws Exception {
    try {
        throw( new MyExc1() );
    }
    catch( MyExc3 h ) {
        System.out.print(2);
    }
    catch( Exception h ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
}
```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 29**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {
    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
}

public static void main(String[] args) {
    long l = 11;
    int i = 2;
    System.out.print(l/i + " ");
    m(l, i);
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

3. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
  - B. `c = (Integer) u;`
  - C. `c = u;`
  - D. `f = c;`
  - E. `u = f;`
- 

4. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

6. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con sf new C.I().

D. Un'istanza della classe I può essere costruita con new C().new I() anche fuori da pk.

E. Nessuna delle precedenti.

---

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        static void q() throws Exception {
            try {
                System.out.print(2);
                throw( new Exception() );
            }
            catch( MyExc2 r ) {
                System.out.print(3);
            }
            catch( MyExc1 c ) {
                System.out.print(4);
            }
            catch( Exception g ) {
                throw( new MyExc1() );
            }
            finally {
                System.out.print(5);
            }
        }
    }
}
```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
    }
}
```

```

        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
}

```

```

    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }

public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        } catch( MyExc1 t ) {
            System.out.print(2);
        } catch( MyExc3 i ) {
            System.out.print(3);
        } finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }

    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        } catch( MyExc3 f ) {
        } catch( MyExc2 c ) {
        }
    }
}

```

```

    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514  
B. 514Exception in thread main MyExc3  
C. 51  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

12. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
}

```

- A. IAC  
B. AAA  
C. CCC  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione
- 

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;  
B. g = (Exception) u;  
C. x = (Error) g;  
D. g = (Exception) x;  
E. Nessuno dei precedenti
- 

14. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}  
B. protected void m(int i, char c) throws RuntimeException {}  
C. public final int m() {return 1;}  
D. public void m(int i){}  
E. Nessuna delle precedenti.
- 

15. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

13. Date le dichiarazioni:

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 
16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
```

```
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 30**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;

- B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1  
B. 2451  
C. 2Exception in thread main MyExc1  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
}

```

```

public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}

```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5  
B. 5 5.5  
C. 5 5.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

7. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;  
B. c = (Integer) u;  
C. c = u;  
D. f = c;  
E. u = f;
- 

8. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

9. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe `I` può essere costruita solo all'interno della classe `C`.
  - B. Un'istanza della classe `I` può essere costruita solo all'interno del pacchetto `pk`.
  - C. All'esterno del pacchetto `pk`, un'istanza della classe `I` può essere costruita con `sf new C.I()`.
  - D. Un'istanza della classe `I` può essere costruita con `new C().new I()` anche fuori da `pk`.
  - E. Nessuna delle precedenti.
- 

10. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}
```

```
abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}
```

```
class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
```

```
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

12. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
    }
}
```

```

        catch( MyExc3 i ) {
    }
    catch( Exception c ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514  
B. 514Exception in thread main MyExc3  
C. 51  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

### 13. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
}

```

```

        c.m(c);
    }
}
A. IAC
B. AAA
C. CCC
D. Errore a tempo di compilazione
E. Errore a tempo di esecuzione

```

---

### 14. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.  
B. Al termine della linea 6.  
C. Al termine della linea 7.  
D. Al termine della linea 8.  
E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

### 15. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c)  
throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

---

16. Quale output si ottiene invocando il metodo m?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
}
```

```
void q(Float f2, char [] a2, Float f5) {  
    if(f2 == f1) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(a1 == a2) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(f5 == f4) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
}
```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 31

1. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. 134

B. 14

C. Errore a tempo di compilazione

D. 14Exception in thread main MyExc1

E. Nessuna delle precedenti

3. Dato un file contenente il seguente codice:

```
package pk;
public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
- D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
- E. Nessuna delle precedenti.

4. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

if(a1 == a2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(f5 == f4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}

```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {

```

```

            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
}

public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 8. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 10. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
- 

#### 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
}

```

```

public static void main(String[] args) {
    long l = 11;
    int i = 2;
    System.out.print(l/i + " ");
    m(l, i);
}

```

- A. 55  
B. 55.5  
C. 55.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

12. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}  
B. protected void m(int i, char c)  
throws RuntimeException {}  
C. public final int m() {return 1;}  
D. public void m(int i){}  
E. Nessuna delle precedenti.
- 

13. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
}

```

```

    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC  
B. AAA  
C. CCC  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione
- 

14. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;  
B. g = (Exception) u;  
C. x = (Error) g;  
D. g = (Exception) x;  
E. Nessuno dei precedenti
- 

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                q();
                System.out.print(1);
            }
        }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
    }
}

```

```

        catch( Exception g ) {
            throw( new MyExc1() );
        }
    finally {
        System.out.print(5);
    }
}

```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

```

```

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 32**

1. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

2. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

4. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}
```

```
abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
```

```

String c = "B";
protected void m(I i) {
    System.out.print(i.c);
}
protected void m(A a) {
    System.out.print(a.c);
}
protected void m(B c) {
    System.out.print(c.c);
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

#### 5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
    }
}

```

```

        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        static void q() throws Exception {
            try {
                System.out.print(2);
                throw( new Exception() );
            }
            catch( MyExc2 r ) {
                System.out.print(3);
            }
            catch( MyExc1 c ) {
                System.out.print(4);
            }
            catch( Exception g ) {
                throw( new MyExc1() );
            }
            finally {
                System.out.print(5);
            }
        }
    }
}

```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

8. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c)
throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

11. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

- 
12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 5 5  
B. 5 5.5  
C. 5 5.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.

- 
13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant{  
    String s = "Lemon";  
}
```

```
public String type() {  
    return s;  
}  
public static void main(String[] args) {  
    Plant a = new Lemon();  
    a.type();  
}
```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

- 
14. Quale output si ottiene invocando il metodo `m`?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
    void q(Float f2, char [] a2, Float f5) {  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a1 == a2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f5 == f4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 101  
B. 000  
C. 110  
D. 010  
E. 100

- 
15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 33**

1. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
- D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
- E. Nessuna delle precedenti.

2. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c)
throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
```

```
throws Exception {
    try {
        System.out.print(1);
        p();
    }
    catch( MyExc2 t ) {
        throw( new MyExc1() );
    }
}
static void p() throws Exception {
    try {
        throw( new MyExc1() );
    }
    catch( MyExc3 h ) {
        System.out.print(2);
    }
    catch( Exception h ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
```

```

    }
    catch( Exception c ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514  
B. 514Exception in thread main MyExc3  
C. 51  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 5. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;  
B. c = (Integer) u;  
C. c = u;  
D. f = c;  
E. u = f;
- 

#### 6. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
    }
}

```

```

        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

A. 101  
B. 000  
C. 110  
D. 010  
E. 100

---

#### 7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
}
public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}
}

```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

8. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

10. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

11. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`

E. Un array non possiede dei membri

---

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

13. Date le dichiarazioni:

```
Exception g;  
Object u;  
Error x;  
g = new Exception();  
x = new Error();  
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

---

14. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
}
```

```
int length;  
  
void add(Tree t) {  
    trees[length++] = t;  
}  
  
void remove() {  
    length--;  
    players[length] = null;  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant{  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
```

```
protected void m(I i) {
    System.out.print(i.c);
}
protected void m(A a) {
    System.out.print(a.c);
}
protected void m(B c) {
    System.out.print(c.c);
}
public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
```

---

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 34**

1. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.
- D. Al termine della linea 8.
- E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

4. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B b) {  
        System.out.print(b.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();  
        c.f();  
        ((A) c).f();  
        ((B) c).f();  
    }  
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

5. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

6. Qual'è l'output di questo codice?

```
String c = "I";  
void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    protected void f() {}  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B b) {  
        System.out.print(b.c);  
    }  
    public static void main(String[] args) {  
        B c = new B();  
        c.m((I) c);  
        c.m((A) c);  
        c.m(c);  
    }  
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

7. Quale delle seguenti non è un sovraccaricamento corretto del metodo `m` nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134  
 B. 14  
 C. Errore a tempo di compilazione  
 D. 14Exception in thread main MyExc1  
 E. Nessuna delle precedenti
- 

#### 9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {

```

```

            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1  
 B. 2451  
 C. 2Exception in thread main MyExc1  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 55  
 B. 5.55  
 C. 5.5.5d  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti.
- 

#### 11. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;  
 B. g = (Exception) u;  
 C. x = (Error) g;

- D. `g = (Exception) x;`  
E. Nessuno dei precedenti

---

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5  
B. 5 5.5  
C. 5 5.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.

---

13. Dato un file contenente il seguente codice:

```
package pk;  
  
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.  
D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.  
E. Nessuna delle precedenti.

---

14. Quale output si ottiene invocando il metodo m?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
    void q(Float f2, char [] a2, Float f5) {  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a1 == a2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f5 == f4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}  
  
A. 101  
B. 000  
C. 110  
D. 010  
E. 100
```

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
}  
public static void main(String[] args) {  
    Plant a = new Lemon();  
}
```

```
    a.type();  
}  
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            m();  
            System.out.print(1);  
        }  
        catch( MyExc1 t ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 i ) {  
        }  
        catch( Exception c ) {  
            System.out.print(3);  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(5);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 f ) {  
        }  
        catch( MyExc2 c ) {  
        }  
        catch( MyExc1 u ) {  
            System.out.print(6);  
        }  
    }  
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 35**

1. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
- D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
- E. Nessuna delle precedenti.

2. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}
```

```
abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

4. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`

E. Un array non possiede dei membri

---

5. Date le dichiarazioni:

```
Exception g;  
Object u;  
Error x;  
g = new Exception();  
x = new Error();  
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 55
  - B. 55.5
  - C. 55.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            m();  
            System.out.print(1);  
        }  
        catch( MyExc1 t ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 i ) {  
        }  
        catch( Exception c ) {  
            System.out.print(3);  
        }  
    }  
}
```

```

        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514  
B. 514Exception in thread main MyExc3  
C. 51  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 55  
B. 55.5  
C. 55.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

11. Date le classi

```

class Tree {
    String type;
    Tree(String s) {

```

```

        type = s;
    }
}

class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.  
B. Al termine della linea 6.  
C. Al termine della linea 7.  
D. Al termine della linea 8.  
E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

12. Qual è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}
```

```

    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)

```

```

        throws Exception {
            try {
                q();
                System.out.print(1);
            }
        }
        static void q() throws Exception {
            try {
                System.out.print(2);
                throw( new Exception() );
            }
            catch( MyExc2 r ) {
                System.out.print(3);
            }
            catch( MyExc1 c ) {
                System.out.print(4);
            }
            catch( Exception g ) {
                throw( new MyExc1() );
            }
            finally {
                System.out.print(5);
            }
        }
    }
}

```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
}

A. 101
B. 000
C. 110
D. 010
E. 100

```

---

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {

```

```

            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 36**

1. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

2. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

```
c.f();
((A) c).f();
((B) c).f();
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA

C. CCC

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

- 
4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

A. 55

B. 55.5

C. 55.5d

D. Errore a tempo di compilazione

E. Nessuna delle precedenti.

- 
5. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
  
    void remove() {  
        length--;  
        players[length] = null;  
    }  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);
```

```
5.         t = null;  
6.         p.remove();  
7.         p = null;  
8.         return t;  
9.     }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

A. Al termine della linea 5.

B. Al termine della linea 6.

C. Al termine della linea 7.

D. Al termine della linea 8.

E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

- 
6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                q();  
                System.out.print(1);  
            }  
            static void q() throws Exception {  
                try {  
                    System.out.print(2);  
                    throw( new Exception() );  
                }  
                catch( MyExc2 r ) {  
                    System.out.print(3);  
                }  
                catch( MyExc1 c ) {  
                    System.out.print(4);  
                }  
                catch( Exception g ) {  
                    throw( new MyExc1() );  
                }  
                finally {  
                    System.out.print(5);  
                }  
            }  
        }
```

A. 2Exception in thread main MyExc1

B. 2451

C. 2Exception in thread main MyExc1

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

- 
7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

8. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

9. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {

```

```

        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101
- B. 000
- C. 110
- D. 010
- E. 100

---

10. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
  - B. `c = (Integer) u;`
  - C. `c = u;`
  - D. `f = c;`
  - E. `u = f;`
- 

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
    }
}

```

```

    }
    catch( MyExc3 h ) {
        System.out.print(2);
    }
    catch( Exception h ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 134  
B. 14  
C. Errore a tempo di compilazione  
D. 14Exception in thread main MyExc1  
E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente  
D. Non tutti i tipi di eccezioni estendono la classe RuntimeException  
E. Un array non possiede dei membri

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

15. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

A. static long m(long l){return l;}  
B. protected void m(int i, char c) throws RuntimeException {}

- C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

```
        throw( new MyExc3() );
    }
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 37**

1. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
```

```
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        static void q() throws Exception {
            try {
                System.out.print(2);
                throw( new Exception() );
            }
            catch( MyExc2 r ) {
                System.out.print(3);
            }
            catch( MyExc1 c ) {
                System.out.print(4);
            }
            catch( Exception g ) {
                throw( new MyExc1() );
            }
            finally {
                System.out.print(5);
            }
        }
    }
}
```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
}
```

```

private void type() {
    System.out.println(s);
}
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree  
B. Lemon  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 
4. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}  
B. protected void m(int i, char c)  
throws RuntimeException {}  
C. public final int m() {return 1;}  
D. public void m(int i){}  
E. Nessuna delle precedenti.
- 
5. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

if(a1 == a2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(f5 == f4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}

```

- A. 101  
B. 000  
C. 110  
D. 010  
E. 100
- 
6. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;  
B. c = (Integer) u;  
C. c = u;  
D. f = c;  
E. u = f;
- 
7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
}

```

```

static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}

```

- A. 514  
B. 514Exception in thread main MyExc3  
C. 51  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 55  
B. 55.5  
C. 55.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }
}

```

```

public static void main(String[] args) {
    long l = 11;
    int i = 2;
    System.out.print(l/i + " ");
    m(l, i);
}

A. 55
B. 55.5
C. 55.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.

```

---

10. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}

class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.  
B. Al termine della linea 6.  
C. Al termine della linea 7.  
D. Al termine della linea 8.  
E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
-

11. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

12. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
- D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
```

```
    System.out.print(3);
}
finally {
    System.out.print(4);
}
}
```

- A. 134  
B. 14  
C. Errore a tempo di compilazione  
D. 14Exception in thread main MyExc1  
E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;  
B. g = (Exception) u;  
C. x = (Error) g;  
D. g = (Exception) x;  
E. Nessuno dei precedenti
- 

16. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente  
D. Non tutti i tipi di eccezioni estendono la classe RuntimeException  
E. Un array non possiede dei membri
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 38**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Dato un file contenente il seguente codice:

```
package pk;

public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
- D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
- E. Nessuna delle precedenti.

3. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

A. static long m(long l){return l;}
B. protected void m(int i, char c)
    throws RuntimeException {}
C. public final int m() {return 1;}
D. public void m(int i){}
E. Nessuna delle precedenti.
```

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
    }
}
```

```

    }
    catch( Exception h ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. 134  
 B. 14  
 C. Errore a tempo di compilazione  
 D. 14Exception in thread main MyExc1  
 E. Nessuna delle precedenti
- 

5. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101  
 B. 000  
 C. 110  
 D. 010  
 E. 100
- 

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;  
 B. g = (Exception) u;  
 C. x = (Error) g;  
 D. g = (Exception) x;  
 E. Nessuno dei precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

A. 55
B. 5.55
C. 5.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.

```

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
}

```

6. Date le dichiarazioni:

```

public String type() {
    return s;
}
public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Qual'è l'output di questo codice?

```

interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}

```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 55
  - B. 55.5
  - C. 55.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1
- B. 2451

C. 2Exception in thread main MyExc1

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

12. Dire quale delle seguenti affermazioni è vera:

A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri

B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente

D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`

E. Un array non possiede dei membri

---

13. Date le dichiarazioni:

```
Integer c;
Object f;
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

A. `u = c;`

B. `c = (Integer) u;`

C. `c = u;`

D. `f = c;`

E. `u = f;`

---

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
    }
}
```

```
finally {
    System.out.print(4);
    throw( new MyExc3() );
}
}
static void m() {
    try {
        System.out.print(5);
        throw( new MyExc3() );
    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}
}
```

A. 514

B. 514Exception in thread main MyExc3

C. 51

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

15. Date le classi

```
class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}
```

e il metodo g

```
1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

16. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
```

```
String c = "B";
protected void m(I i) {
    System.out.print(i.c);
}
protected void m(A a) {
    System.out.print(a.c);
}
protected void m(B b) {
    System.out.print(b.c);
}
public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
    c.m(c);
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 39**

1. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {
    protected int m(int x) throws Exception {
        return x;
    }
}
```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c)
throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

2. Qual'è l'output di questo codice?

```
String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        static void q() throws Exception {
            try {
                System.out.print(2);
                throw( new Exception() );
            }
            catch( MyExc2 r ) {
                System.out.print(3);
            }
            catch( MyExc1 c ) {
                System.out.print(4);
            }
            catch( Exception g ) {
                throw( new MyExc1() );
            }
            finally {
                System.out.print(5);
            }
        }
    }
}
```

- A. 25Exception in thread main MyExc1
  - B. 2451
  - C. 2Exception in thread main MyExc1
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}
```

- A. 514
- B. 514Exception in thread main MyExc3
- C. 51
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}
abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
```

```
        System.out.println(s);
    }
}
class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

8. Quale output si ottiene invocando il metodo m?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
    void q(Float f2, char [] a2, Float f5) {  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a1 == a2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f5 == f4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

9. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
  
    void remove() {  
        length--;  
        trees[length] = null;  
    }  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

11. Qual'è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B c) {  
        System.out.print(c.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();  
        c.f();  
        ((A) c).f();  
        ((B) c).f();  
    }  
}
```

- A. IAC
  - B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

12. Dato un file contenente il seguente codice:

```
package pk;  
  
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

13. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `u = c;`
  - B. `c = (Integer) u;`
  - C. `c = u;`
  - D. `f = c;`
  - E. `u = f;`
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    void type() {  
        System.out.println(s);  
    }  
}  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
- B. Lemon

- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args) {  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 5 5
- B. 5 5.5

- C. 5 5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

16. Date le dichiarazioni:

```
Exception g;  
Object u;  
Error x;  
g = new Exception();  
x = new Error();  
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 40**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
}

public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
```

```
} // Linea vuota

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
}

public static void main(String[] args) {
    I c = new B();
    c.f();
    ((A) c).f();
    ((B) c).f();
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }

public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
    }
}
```

```

        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 25Exception in thread main MyExc1  
B. 2451  
C. 2Exception in thread main MyExc1  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args) {
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 55  
B. 55.5  
C. 55.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

5. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
    }
}

```

```

        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 101  
B. 000  
C. 110  
D. 010  
E. 100
- 

6. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 134
  - B. 14
  - C. Errore a tempo di compilazione
  - D. 14Exception in thread main MyExc1
  - E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public void m(long l, int i) {  
        float f = l/i;  
        System.out.print(f);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        int i = 2;  
        System.out.print(l/i + " ");  
        m(l, i);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

---

8. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe RuntimeException
  - E. Un array non possiede dei membri
- 

9. Date le dichiarazioni:

```
Exception g;  
Object u;  
Error x;  
g = new Exception();  
x = new Error();  
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

10. Dato un file contenente il seguente codice:

```
package pk;  
  
public class C {  
    static class I {  
    }  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con sf new C.I().
  - D. Un'istanza della classe I può essere costruita con new C().new I() anche fuori da pk.
  - E. Nessuna delle precedenti.
- 

11. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
- 

12. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c) throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

### 13. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;

    void add(Tree t) {
        trees[length++] = t;
    }

    void remove() {
        length--;
        players[length] = null;
    }
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

### 14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

A. Tree
B. Lemon
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

### 15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
    }
}

```

```

    }
    catch( MyExc3 f ) {
    }
    catch( MyExc2 c ) {
    }
    catch( MyExc1 u ) {
        System.out.print(6);
    }
}

```

- A. 514  
 B. 514Exception in thread main MyExc3  
 C. 51  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

16. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

```

```

    }

    class B extends A {
        String c = "B";
        protected void m(I i) {
            System.out.print(i.c);
        }
        protected void m(A a) {
            System.out.print(a.c);
        }
        protected void m(B b) {
            System.out.print(b.c);
        }
    }

    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

- A. IAC  
 B. AAA  
 C. CCC  
 D. Errore a tempo di compilazione  
 E. Errore a tempo di esecuzione
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 41**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
    }
}
```

```
        System.out.print(l/i + " ");
        m(l, i);
    }
}
```

- A. 55
- B. 55.5
- C. 55.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1

E. Nessuna delle precedenti

---

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

A. 25Exception in thread main MyExc1

B. 2451

C. 2Exception in thread main MyExc1

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

5. Quale output si ottiene invocando il metodo m?

```
class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        }
    }
}
```

```
    } else {
        System.out.print(0);
    }
    if(a1 == a2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f5 == f4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 101
  - B. 000
  - C. 110
  - D. 010
  - E. 100
- 

6. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC

- B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

## 7. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;  
  
    void add(Tree t) {  
        trees[length++] = t;  
    }  
  
    void remove() {  
        length--;  
        players[length] = null;  
    }  
}
```

e il metodo g

```
1. Tree g(String s) {  
2.     Park p = new Park();  
3.     Tree t = new Tree(s);  
4.     p.add(t);  
5.     t = null;  
6.     p.remove();  
7.     p = null;  
8.     return t;  
9. }
```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

## 8. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- 

- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
  - D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
  - E. Un array non possiede dei membri
- 

## 9. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            m();  
            System.out.print(1);  
        }  
        catch( MyExc1 t ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 i ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc3() );  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(5);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 f ) {  
        }  
        catch( MyExc2 c ) {  
        }  
        catch( MyExc1 u ) {  
            System.out.print(6);  
        }  
    }  
}
```

- A. 514
  - B. 514Exception in thread main MyExc3
  - C. 51
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

## 10. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```

class C {
    protected int m(int x) throws Exception {
        return x;
    }
}

```

- A. static long m(long l){return l;}
  - B. protected void m(int i, char c) throws RuntimeException {}
  - C. public final int m() {return 1;}
  - D. public void m(int i){}
  - E. Nessuna delle precedenti.
- 

11. Date le dichiarazioni:

```

Integer c;
Object f;
String u;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
  - B. c = (Integer) u;
  - C. c = u;
  - D. f = c;
  - E. u = f;
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public static void m(long l, char c) {
        double d = l/c;
        System.out.print(d);
    }

    public static void main(String[] args){
        long l = 11;
        char c = 2;
        System.out.print(l/c + " ");
        m(l, c);
    }
}

```

- A. 5 5
  - B. 5 5.5
  - C. 5 5.5d
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Date le dichiarazioni:

```

Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
  - B. g = (Exception) u;
  - C. x = (Error) g;
  - D. g = (Exception) x;
  - E. Nessuno dei precedenti
- 

15. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
    public static void main(String[] args) {
        B c = new B();
        c.m((I) c);
        c.m((A) c);
        c.m(c);
    }
}

```

A. IAC

- B. AAA
  - C. CCC
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

16. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `sf new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 42**

1. Date le dichiarazioni:

```
Exception g;
Object u;
Error x;
g = new Exception();
x = new Error();
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

2. Qual'è l'output di questo codice?

```
interface I {
    String c = "I";
    void f();
}

abstract class A implements I {
    String c = "A";
    public void f() {
        System.out.print(c);
    }
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B c) {
        System.out.print(c.c);
    }
    public static void main(String[] args) {
        I c = new B();
```

```
        c.f();
        ((A) c).f();
        ((B) c).f();
    }
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(2);
            throw( new Exception() );
        }
        catch( MyExc2 r ) {
            System.out.print(3);
        }
        catch( MyExc1 c ) {
            System.out.print(4);
        }
        catch( Exception g ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 25Exception in thread main MyExc1
- B. 2451

C. 2Exception in thread main MyExc1

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

A. 55

B. 55.5

C. 55.5d

D. Errore a tempo di compilazione

E. Nessuna delle precedenti.

---

5. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

A. static long m(long l){return l;}

B. protected void m(int i, char c)  
throws RuntimeException {}

C. public final int m() {return 1;}

D. public void m(int i){}

E. Nessuna delle precedenti.

---

6. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

A. u = c;

B. c = (Integer) u;

C. c = u;

D. f = c;

E. u = f;

---

7. Quale output si ottiene invocando il metodo m?

```
class B {  
    private Float f1 = new Float(0.5);  
    private Float f3 = f1;  
    private char [] a1 = new char [5];  
    private Float f4 = new Float(20.0);  
    void m() {  
        char [] a3;  
        a3 = new char [5];  
        q(f3, a3, f4);  
    }  
    void q(Float f2, char [] a2, Float f5) {  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(a1 == a2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f5 == f4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

A. 101

B. 000

C. 110

D. 010

E. 100

---

8. Date le classi

```
class Tree {  
    String type;  
    Tree(String s) {  
        type = s;  
    }  
}  
class Park {  
    private Tree[] trees = new Tree[100];  
    int length;
```

```

void add(Tree t) {
    trees[length++] = t;
}

void remove() {
    length--;
    players[length] = null;
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 6.
  - C. Al termine della linea 7.
  - D. Al termine della linea 8.
  - E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    private void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant {
    String s = "Lemon";
    public String type() {
        return s;
    }
}

public static void main(String[] args) {
    Plant a = new Lemon();
    a.type();
}

```

- A. Tree
  - B. Lemon
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

A. 5.5
B. 5.5.5
C. 5.5d
D. Errore a tempo di compilazione
E. Nessuna delle precedenti.

```

---

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( MyExc2 t ) {
            throw( new MyExc1() );
        }
    }
    static void p() throws Exception {
        try {
            throw( new MyExc1() );
        }
        catch( MyExc3 h ) {
            System.out.print(2);
        }
        catch( Exception h ) {
            System.out.print(3);
        }
    }
}

```

```

        }
    finally {
        System.out.print(4);
    }
}

```

- A. 134  
 B. 14  
 C. Errore a tempo di compilazione  
 D. 14Exception in thread main MyExc1  
 E. Nessuna delle precedenti
- 

12. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
 C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente  
 D. Non tutti i tipi di eccezioni estendono la classe RuntimeException  
 E. Un array non possiede dei membri
- 

13. Qual'è l'output di questo codice?

```

String c = "I";
void f();
}

abstract class A implements I {
    String c = "A";
    protected void f() {}
    abstract void m(A a);
}

class B extends A {
    String c = "B";
    protected void m(I i) {
        System.out.print(i.c);
    }
    protected void m(A a) {
        System.out.print(a.c);
    }
    protected void m(B b) {
        System.out.print(b.c);
    }
}

public static void main(String[] args) {
    B c = new B();
    c.m((I) c);
    c.m((A) c);
}

```

```

        c.m(c);
    }
}

```

- A. IAC  
 B. AAA  
 C. CCC  
 D. Errore a tempo di compilazione  
 E. Errore a tempo di esecuzione
- 

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
            System.out.print(6);
        }
    }
}

```

- A. 514  
 B. 514Exception in thread main MyExc3  
 C. 51  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public interface Plant {
    String s = "Plant";
    String type();
}

abstract class Tree {
    String s = "Tree";
    Tree() {
        type();
    }
    void type() {
        System.out.println(s);
    }
}

class Lemon extends Tree implements Plant{
    String s = "Lemon";
    public String type() {
        return s;
    }
    public static void main(String[] args) {
        Plant a = new Lemon();
        a.type();
    }
}

```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

16. Dato un file contenente il seguente codice:

```

package pk;

public class C {
    static class I {
    }
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `sf new C.I()`.
  - D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 43

1. Date le dichiarazioni:

```
Integer c;  
Object f;  
String u;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. u = c;
- B. c = (Integer) u;
- C. c = u;
- D. f = c;
- E. u = f;

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( MyExc2 t ) {  
                throw( new MyExc1() );  
            }  
            static void p() throws Exception {  
                try {  
                    throw( new MyExc1() );  
                }  
                catch( MyExc3 h ) {  
                    System.out.print(2);  
                }  
                catch( Exception h ) {  
                    System.out.print(3);  
                }  
                finally {  
                    System.out.print(4);  
                }  
            }  
        }  
}
```

- A. 134
- B. 14
- C. Errore a tempo di compilazione
- D. 14Exception in thread main MyExc1
- E. Nessuna delle precedenti

3. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                q();  
                System.out.print(1);  
            }  
            static void q() throws Exception {  
                try {  
                    System.out.print(2);  
                    throw( new Exception() );  
                }  
                catch( MyExc2 r ) {  
                    System.out.print(3);  
                }  
                catch( MyExc1 c ) {  
                    System.out.print(4);  
                }  
                catch( Exception g ) {  
                    throw( new MyExc1() );  
                }  
                finally {  
                    System.out.print(5);  
                }  
            }  
        }  
}
```

- A. 25Exception in thread main MyExc1
- B. 2451
- C. 2Exception in thread main MyExc1
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class A {  
  
    public static void m(long l, char c) {  
        double d = l/c;  
        System.out.print(d);  
    }  
  
    public static void main(String[] args){  
        long l = 11;  
        char c = 2;  
        System.out.print(l/c + " ");  
        m(l, c);  
    }  
}
```

- A. 5 5
- B. 5 5.5
- C. 5 5.5d
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti.

- 
5. Dire quale delle seguenti affermazioni è vera:

- A. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- B. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- D. Non tutti i tipi di eccezioni estendono la classe `RuntimeException`
- E. Un array non possiede dei membri

- 
6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    void type() {  
        System.out.println(s);  
    }  
}
```

```
class Lemon extends Tree implements Plant{  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- 
7. Qual è l'output di questo codice?

```
interface I {  
    String c = "I";  
    void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    public void f() {  
        System.out.print(c);  
    }  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B c) {  
        System.out.print(c.c);  
    }  
    public static void main(String[] args) {  
        I c = new B();  
        c.f();  
        ((A) c).f();  
        ((B) c).f();  
    }  
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public interface Plant {  
    String s = "Plant";  
    String type();  
}  
  
abstract class Tree {  
    String s = "Tree";  
    Tree() {  
        type();  
    }  
    private void type() {  
        System.out.println(s);  
    }  
}  
  
class Lemon extends Tree implements Plant {  
    String s = "Lemon";  
    public String type() {  
        return s;  
    }  
    public static void main(String[] args) {  
        Plant a = new Lemon();  
        a.type();  
    }  
}
```

- A. Tree
- B. Lemon
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Date le dichiarazioni:

```
Exception g;  
Object u;  
Error x;  
g = new Exception();  
x = new Error();  
u = new Error();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. x = (Error) u;
- B. g = (Exception) u;
- C. x = (Error) g;
- D. g = (Exception) x;
- E. Nessuno dei precedenti

10. Quale delle seguenti non è un sovraccaricamento corretto del metodo m nella classe C?

```
class C {  
    protected int m(int x) throws Exception {  
        return x;  
    }  
}
```

- A. static long m(long l){return l;}
- B. protected void m(int i, char c) throws RuntimeException {}
- C. public final int m() {return 1;}
- D. public void m(int i){}
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo codice?

```
String c = "I";  
void f();  
}  
  
abstract class A implements I {  
    String c = "A";  
    protected void f() {}  
    abstract void m(A a);  
}  
  
class B extends A {  
    String c = "B";  
    protected void m(I i) {  
        System.out.print(i.c);  
    }  
    protected void m(A a) {  
        System.out.print(a.c);  
    }  
    protected void m(B b) {  
        System.out.print(b.c);  
    }  
    public static void main(String[] args) {  
        B c = new B();  
        c.m((I) c);  
        c.m((A) c);  
        c.m(c);  
    }  
}
```

- A. IAC
- B. AAA
- C. CCC
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

12. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            m();
            System.out.print(1);
        }
        catch( MyExc1 t ) {
            System.out.print(2);
        }
        catch( MyExc3 i ) {
        }
        catch( Exception c ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc3() );
        }
    }
    static void m() {
        try {
            System.out.print(5);
            throw( new MyExc3() );
        }
        catch( MyExc3 f ) {
        }
        catch( MyExc2 c ) {
        }
        catch( MyExc1 u ) {
            System.out.print(6);
        }
    }
}

```

- A. 514  
B. 514Exception in thread main MyExc3  
C. 51  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class A {

    public void m(long l, int i) {
        float f = l/i;
        System.out.print(f);
    }

    public static void main(String[] args) {
        long l = 11;
        int i = 2;
        System.out.print(l/i + " ");
        m(l, i);
    }
}

```

- A. 55  
B. 55.5  
C. 55.5d  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti.
- 

14. Quale output si ottiene invocando il metodo m?

```

class B {
    private Float f1 = new Float(0.5);
    private Float f3 = f1;
    private char [] a1 = new char [5];
    private Float f4 = new Float(20.0);
    void m() {
        char [] a3;
        a3 = new char [5];
        q(f3, a3, f4);
    }
    void q(Float f2, char [] a2, Float f5) {
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(a1 == a2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f5 == f4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

A. 101  
B. 000  
C. 110  
D. 010  
E. 100

---

15. Date le classi

```

class Tree {
    String type;
    Tree(String s) {
        type = s;
    }
}
class Park {
    private Tree[] trees = new Tree[100];
    int length;
}

```

```

void add(Tree t) {
    trees[length++] = t;
}

void remove() {
    length--;
    players[length] = null;
}

```

e il metodo g

```

1. Tree g(String s) {
2.     Park p = new Park();
3.     Tree t = new Tree(s);
4.     p.add(t);
5.     t = null;
6.     p.remove();
7.     p = null;
8.     return t;
9. }

```

Quando l'oggetto creato alla linea 3 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 6.
- C. Al termine della linea 7.

D. Al termine della linea 8.

E. Non si può dire, poiché l'oggetto viene restituito dal metodo g.

16. Dato un file contenente il seguente codice:

```
package pk;
```

```
public class C {
    static class I {
    }
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe C.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Allesterno del pacchetto pk, un'istanza della classe I può essere costruita con `new C.I()`.
- D. Un'istanza della classe I può essere costruita con `new C().new I()` anche fuori da pk.
- E. Nessuna delle precedenti.

**Prova n. 1**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 2**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 3**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 4**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 5**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 6**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 8**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 9**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 10**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 11**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 12**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 16**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

8 settembre 2016

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 1:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 2:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 3:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 4:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 5:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 6:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  | ■ | ■ |   |   |   |   |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  |   |   |   |   | ■ | ■ |   |
| 6  | ■ | ■ |   |   |   |   |   |
| 7  |   |   |   |   | ■ | ■ |   |
| 8  |   |   |   |   | ■ | ■ |   |
| 9  | ■ | ■ |   |   |   |   |   |
| 10 |   |   | ■ | ■ |   |   |   |
| 11 |   |   |   |   | ■ | ■ |   |
| 12 |   |   |   |   | ■ | ■ |   |
| 13 |   |   | ■ | ■ |   |   |   |
| 14 |   |   |   |   | ■ | ■ |   |
| 15 |   |   |   | ■ |   |   |   |
| 16 |   |   |   |   |   | ■ | ■ |

Risultato prova n. 7:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 8:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 9:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 10:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 11:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   | ■ |   |   |   |
| 3  |   |   | ■ | ■ |   |   |   |
| 4  |   |   |   | ■ |   |   |   |
| 5  |   |   |   |   | ■ | ■ |   |
| 6  |   |   |   |   |   | ■ |   |
| 7  |   |   |   |   |   | ■ |   |
| 8  |   |   |   |   |   | ■ |   |
| 9  |   |   |   |   |   |   | ■ |
| 10 |   |   |   |   |   | ■ | ■ |
| 11 |   |   |   |   |   | ■ | ■ |
| 12 | ■ |   | ■ |   |   |   |   |
| 13 |   |   |   |   | ■ |   |   |
| 14 |   | ■ | ■ |   |   |   |   |
| 15 |   | ■ | ■ |   |   |   |   |
| 16 |   |   | ■ |   |   |   |   |

Risultato prova n. 12:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 13:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 14:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 15:

|    |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|--|
| 1  |   | ■ |   |   |   |   |  |
| 2  |   |   |   |   | ■ | ■ |  |
| 3  |   |   | ■ | ■ |   |   |  |
| 4  | ■ | ■ |   |   |   |   |  |
| 5  | ■ | ■ |   |   |   |   |  |
| 6  |   |   | ■ | ■ |   |   |  |
| 7  |   |   |   | ■ | ■ |   |  |
| 8  |   |   |   |   | ■ | ■ |  |
| 9  | ■ | ■ |   |   |   |   |  |
| 10 |   |   |   |   | ■ | ■ |  |
| 11 |   |   |   |   | ■ | ■ |  |
| 12 |   |   |   |   |   |   |  |
| 13 |   |   |   |   | ■ | ■ |  |
| 14 |   |   |   |   | ■ | ■ |  |
| 15 |   |   | ■ | ■ |   |   |  |
| 16 |   |   |   |   |   | ■ |  |

Risultato prova n. 16:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  | ■ | ■ |   |   |   |   |   |
| 4  |   |   |   |   | ■ | ■ |   |
| 5  |   |   | ■ | ■ |   |   |   |
| 6  |   |   |   |   | ■ | ■ |   |
| 7  | ■ | ■ |   |   |   |   |   |
| 8  | ■ | ■ |   |   |   |   |   |
| 9  | ■ | ■ |   |   |   |   |   |
| 10 |   |   | ■ | ■ |   |   |   |
| 11 |   |   |   | ■ | ■ |   |   |
| 12 |   |   |   |   | ■ | ■ |   |
| 13 |   |   |   |   |   | ■ | ■ |
| 14 |   |   |   |   | ■ | ■ |   |
| 15 |   |   |   |   | ■ | ■ |   |
| 16 |   |   |   |   | ■ | ■ |   |

Risultato prova n. 17:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   | ■ |   |   |   |   |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  |   |   |   |   |   | ■ | ■ |
| 5  |   |   |   |   | ■ | ■ |   |
| 6  |   |   | ■ | ■ |   |   |   |
| 7  |   |   | ■ | ■ |   |   |   |
| 8  | ■ | ■ |   |   |   |   |   |
| 9  |   |   |   | ■ | ■ |   |   |
| 10 |   |   |   | ■ | ■ |   |   |
| 11 | ■ | ■ |   |   |   |   |   |
| 12 | ■ | ■ |   |   |   |   |   |
| 13 |   |   |   | ■ | ■ |   |   |
| 14 |   |   |   |   | ■ | ■ |   |
| 15 |   |   |   |   |   | ■ | ■ |
| 16 |   |   |   |   |   | ■ | ■ |

Risultato prova n. 18:

|    |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|--|
| 1  |   | ■ |   |   |   |   |  |
| 2  |   |   |   |   |   | ■ |  |
| 3  |   |   |   | ■ | ■ |   |  |
| 4  |   |   | ■ | ■ |   |   |  |
| 5  |   |   | ■ | ■ |   |   |  |
| 6  | ■ | ■ |   |   |   |   |  |
| 7  |   |   |   |   | ■ | ■ |  |
| 8  |   |   |   |   | ■ | ■ |  |
| 9  |   |   |   |   | ■ | ■ |  |
| 10 |   |   |   |   | ■ | ■ |  |
| 11 |   |   |   |   | ■ | ■ |  |
| 12 | ■ | ■ |   |   |   |   |  |
| 13 |   |   |   |   | ■ | ■ |  |
| 14 |   |   | ■ | ■ |   |   |  |
| 15 | ■ | ■ |   |   |   |   |  |
| 16 |   |   |   |   | ■ | ■ |  |

Risultato prova n. 19:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 20:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 21:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  | ■ | ■ |   |   |   |   |   |
| 3  |   |   |   |   | ■ | ■ |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  |   |   | ■ | ■ |   |   |   |
| 6  |   |   |   |   | ■ | ■ |   |
| 7  |   |   |   |   |   | ■ | ■ |
| 8  |   |   |   |   | ■ | ■ |   |
| 9  |   |   |   |   | ■ | ■ |   |
| 10 |   |   |   |   | ■ | ■ |   |
| 11 |   |   |   |   | ■ | ■ |   |
| 12 |   |   |   |   | ■ | ■ |   |
| 13 |   |   |   | ■ | ■ |   |   |
| 14 | ■ | ■ |   |   |   |   |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 |   |   |   | ■ |   |   |   |

Risultato prova n. 22:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 23:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 24:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 25:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  |   |   | ■ | ■ |   |   |   |
| 4  |   |   |   | ■ |   |   |   |
| 5  |   |   |   |   | ■ | ■ |   |
| 6  |   |   |   |   |   | ■ | ■ |
| 7  |   |   |   |   | ■ | ■ |   |
| 8  |   |   |   |   | ■ | ■ |   |
| 9  |   |   | ■ | ■ |   |   |   |
| 10 | ■ | ■ |   |   |   |   |   |
| 11 |   |   |   |   | ■ | ■ |   |
| 12 |   |   |   | ■ | ■ |   |   |
| 13 |   |   |   |   | ■ | ■ |   |
| 14 | ■ | ■ |   |   |   |   |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 |   |   |   |   | ■ | ■ |   |

Risultato prova n. 26:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 27:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  |   |   |   |   | ■ | ■ |   |
| 4  |   |   |   |   | ■ | ■ |   |
| 5  | ■ | ■ |   |   |   |   |   |
| 6  |   |   |   | ■ | ■ |   |   |
| 7  |   |   |   |   | ■ | ■ |   |
| 8  | ■ | ■ |   |   |   |   |   |
| 9  | ■ | ■ |   |   |   |   |   |
| 10 |   |   |   |   | ■ | ■ |   |
| 11 |   |   |   |   | ■ | ■ |   |
| 12 |   |   | ■ | ■ |   |   |   |
| 13 |   |   |   |   | ■ | ■ |   |
| 14 |   |   |   |   |   | ■ | ■ |
| 15 |   |   | ■ | ■ |   |   |   |
| 16 | ■ | ■ |   |   |   |   |   |

Risultato prova n. 28:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 29:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 30:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 31:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 32:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  | ■ | ■ |   |   |   |   |   |
| 4  |   |   |   |   | ■ | ■ |   |
| 5  |   |   |   |   | ■ | ■ |   |
| 6  | ■ | ■ |   |   |   |   |   |
| 7  | ■ | ■ |   |   |   |   |   |
| 8  |   |   |   |   | ■ | ■ |   |
| 9  |   |   |   |   |   | ■ | ■ |
| 10 |   |   |   |   | ■ | ■ |   |
| 11 |   |   |   |   | ■ | ■ |   |
| 12 |   |   |   |   | ■ | ■ |   |
| 13 | ■ | ■ |   |   |   |   |   |
| 14 |   |   | ■ | ■ |   |   |   |
| 15 |   |   |   | ■ | ■ |   |   |
| 16 |   | ■ | ■ |   |   |   |   |

Risultato prova n. 33:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 34:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  |   |   | ■ |   |   |   |   |
| 4  |   |   |   |   | ■ | ■ |   |
| 5  | ■ | ■ |   |   |   |   |   |
| 6  | ■ | ■ |   |   |   |   |   |
| 7  |   |   |   |   | ■ | ■ |   |
| 8  |   |   |   |   |   | ■ | ■ |
| 9  |   |   |   |   | ■ | ■ |   |
| 10 |   |   |   |   | ■ | ■ |   |
| 11 |   |   | ■ |   |   |   |   |
| 12 |   |   |   |   | ■ | ■ |   |
| 13 |   |   |   | ■ | ■ |   |   |
| 14 |   |   |   |   | ■ | ■ |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 | ■ | ■ |   |   |   |   |   |

Risultato prova n. 35:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   | ■ |   |   |   |   |
| 3  |   |   |   |   | ■ | ■ |   |
| 4  |   |   |   |   | ■ | ■ |   |
| 5  |   |   | ■ |   |   |   |   |
| 6  |   |   |   |   | ■ | ■ |   |
| 7  |   |   |   |   |   | ■ | ■ |
| 8  |   |   | ■ |   |   |   |   |
| 9  | ■ | ■ |   |   |   |   |   |
| 10 |   |   |   |   | ■ | ■ |   |
| 11 | ■ | ■ |   |   |   |   |   |
| 12 | ■ | ■ |   |   |   |   |   |
| 13 |   |   |   |   | ■ | ■ |   |
| 14 |   |   |   | ■ | ■ |   |   |
| 15 |   |   |   |   | ■ | ■ |   |
| 16 |   |   |   |   | ■ | ■ |   |

Risultato prova n. 36:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  |   |   | ■ |   |   |
| 2  |   |   |   | ■ |   |
| 3  | ■ |   |   |   |   |
| 4  |   |   |   | ■ |   |
| 5  | ■ |   |   |   |   |
| 6  |   |   |   | ■ |   |
| 7  |   |   |   | ■ |   |
| 8  |   |   |   |   | ■ |
| 9  |   |   |   | ■ |   |
| 10 |   | ■ |   |   |   |
| 11 |   |   |   | ■ |   |
| 12 |   | ■ |   |   |   |
| 13 |   |   | ■ |   |   |
| 14 | ■ | ■ |   |   |   |
| 15 | ■ | ■ |   |   |   |
| 16 |   |   |   | ■ |   |

Risultato prova n. 37:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   | ■ |   |   |   |   |
| 3  |   |   |   |   | ■ | ■ |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  | ■ | ■ |   |   |   |   |   |
| 6  |   | ■ | ■ |   |   |   |   |
| 7  |   |   |   |   |   | ■ | ■ |
| 8  |   |   |   | ■ | ■ |   |   |
| 9  |   |   | ■ | ■ |   |   |   |
| 10 |   |   |   |   | ■ | ■ |   |
| 11 |   |   |   |   | ■ | ■ |   |
| 12 |   |   |   |   | ■ | ■ |   |
| 13 |   |   |   |   | ■ | ■ |   |
| 14 |   |   |   |   | ■ | ■ |   |
| 15 |   |   | ■ | ■ |   |   |   |
| 16 |   |   |   |   | ■ | ■ |   |

Risultato prova n. 38:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 39:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 40:

|    |  |   |   |   |   |   |  |
|----|--|---|---|---|---|---|--|
| 1  |  | ■ |   |   |   |   |  |
| 2  |  |   |   |   | ■ | ■ |  |
| 3  |  | ■ |   |   |   |   |  |
| 4  |  |   |   |   | ■ | ■ |  |
| 5  |  | ■ |   |   |   |   |  |
| 6  |  |   | ■ | ■ |   |   |  |
| 7  |  |   | ■ | ■ |   |   |  |
| 8  |  |   |   |   | ■ | ■ |  |
| 9  |  |   |   |   | ■ | ■ |  |
| 10 |  |   |   |   | ■ | ■ |  |
| 11 |  |   |   |   | ■ | ■ |  |
| 12 |  |   |   |   |   | ■ |  |
| 13 |  |   |   | ■ | ■ |   |  |
| 14 |  | ■ |   |   |   |   |  |
| 15 |  |   |   |   | ■ | ■ |  |
| 16 |  |   | ■ |   |   |   |  |

Risultato prova n. 41:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   | ■ |   |   |   |   |
| 3  |  |   |   | ■ | ■ |   |   |
| 4  |  |   |   |   |   | ■ | ■ |
| 5  |  |   |   |   | ■ | ■ |   |
| 6  |  |   |   |   | ■ | ■ |   |
| 7  |  | ■ | ■ |   |   |   |   |
| 8  |  |   | ■ | ■ |   |   |   |
| 9  |  | ■ | ■ |   |   |   |   |
| 10 |  |   |   | ■ | ■ |   |   |
| 11 |  | ■ | ■ |   |   |   |   |
| 12 |  |   |   | ■ | ■ |   |   |
| 13 |  |   |   | ■ | ■ |   |   |
| 14 |  |   |   | ■ | ■ |   |   |
| 15 |  |   |   | ■ | ■ |   |   |
| 16 |  |   | ■ | ■ |   |   |   |

Risultato prova n. 42:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 43:



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 1**

1. Date le dichiarazioni:

```
Boolean [] c;
Object [] [] s;
Object [] w;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

2. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

3. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
```

```
5.             this.g(o);
6.         }
7.         void g(Object obj) {
8.             obj = null;
9.             obj = new Object();
10.        }
11.        public static void main(String[] a) {
12.            new X().f();
13.        }
14.    }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
    }
}
```

```

        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 423Exception in thread "main" MyExc3  
B. 4222222... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 42Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;  
B. s = (C2) x;  
C. c = (B2) s;  
D. s = (C2) c;  
E. Nessuno dei precedenti
- 

6. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0  
B. 1  
C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

7. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
           i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 000  
B. 101  
C. 100  
D. 011  
E. 001
- 

8. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

```

A. 0

B. 1

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}  
  
public class B extends A {  
    B() {  
        super( 2 );  
    }  
    public static void main( String [] a) {  
        new B();  
    }  
}
```

A. 0

B. 1

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args) {  
        A a = new B();  
        a.m();  
    }  
}
```

A. A.f

B. B.m

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class D1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            q();  
        }  
        static void q() throws Exception {  
            try {  
                throw( new Exception() );  
            }  
            catch( MyExc2 d ) {  
                System.out.print(1);  
                throw( new MyExc3() );  
            }  
            catch( Exception h ) {  
                throw( new MyExc2() );  
            }  
            catch( MyExc3 w ) {  
            }  
            finally {  
                System.out.print(2);  
            }  
        }  
    }  
}
```

A. 111... (ciclo infinito)

B. Errore a tempo di compilazione

C. Exception in thread "main" MyExc2

D. 2Exception in thread "main" MyExc2

E. Nessuna delle precedenti

12. Dire quale delle seguenti affermazioni è vera:

A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

B. Ai metodi static non si applica il *dynamic method dispatch*

C. Una classe non può essere dichiarata final

D. Un array vuoto può avere riferimento null

E. L'operatore new non restituisce un riferimento all'oggetto appena creato

13. Qual è il *sovvraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}  
  
A. public A(String b) {}  
B. void A() {}  
C. private A(int i, String s) {}  
D. public Object A(int i, String s) {}  
E. Il costruttore di una classe non può essere sovraccaricato.
```

14. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
abstract class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args) {  
        I i = new B();  
        i.m();  
    }  
}
```

- A. A.f
- B. B.m
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

15. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

16. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            q();  
            System.out.print(1);  
        }  
        catch( MyExc2 h ) {  
            System.out.print(2);  
        }  
    }  
    static void q() throws Exception {  
        try {  
            throw( new MyExc2() );  
        }  
        catch( Exception w ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 2**

1. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - B. Ai metodi *static* non si applica il *dynamic method dispatch*
  - C. Una classe non può essere dichiarata *final*
  - D. Un array vuoto può avere riferimento *null*
  - E. L'operatore *new* non restituisce un riferimento all'oggetto appena creato
- 

2. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

3. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}
```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
- 

4. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

5. Data la classe

```

1.   class X {
2.       private Object o;
3.       void f() {
4.           o = new Object();
5.           this.g(o);
6.       }
7.       void g(Object obj) {
8.           obj = null;
9.           obj = new Object();
10.      }
11.      public static void main(String[] a) {
12.          new X().f();
13.      }
14.  }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 8.
  - C. Al termine della linea 9.
  - D. Al termine della linea 12.
  - E. Non si può dire.
- 

#### 6. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;
  - B. s = w;
  - C. c = w;
  - D. s = (Object [] []) c;
  - E. w = (Object []) s;
- 

#### 7. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
  - B. s = (C2) x;
  - C. c = (B2) s;
  - D. s = (C2) c;
  - E. Nessuno dei precedenti
- 

#### 8. Qual è il *sovraff장imento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

#### 9. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

abstract class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}

```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

#### 11. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

#### 12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        } catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        } finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        } catch( MyExc3 a ) {
        } finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 423Exception in thread "main" MyExc3
  - B. 4222222... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

#### 13. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        A a = new B();
        a.m();
    }
}

```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 42
  - C. 341
  - D. 41
  - E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
```

```
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
        finally {
            System.out.print(2);
        }
    }
}
```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 3**

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
           i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

2. Qual è l'output di questo codice?

```
public interface I {
    void m();
}

class A implements I {
    public void f() {
```

```
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        A a = new B();
        a.m();
    }
}

A. A.f
B. B.m
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti
```

3. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi static non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata final
- D. Un array vuoto può avere riferimento null
- E. L'operatore new non restituisce un riferimento all'oggetto appena creato

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
}
```

```

static void q() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc2 d ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception h ) {
        throw( new MyExc2() );
    }
    catch( MyExc3 w ) {
    }
    finally {
        System.out.print(2);
    }
}

```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

#### 5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 42
  - C. 341
- 

D. 41

E. Nessuna delle precedenti

---

#### 6. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

A. 0
B. 1
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

```

---

#### 7. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;
  - B. s = w;
  - C. c = w;
  - D. s = (Object [] []) c;
  - E. w = (Object []) s;
- 

#### 8. Se nella classe C è dichiarato il metodo

```

int f(String s)

```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
-

9. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
abstract class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args) {  
        I i = new B();  
        i.m();  
    }  
}
```

- A. A.f
- B. B.m
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

10. Qual è il *sovraffaccimento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class D1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
            System.out.print(1);  
        } catch( MyExc3 r ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        } finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        } catch( MyExc3 a ) {  
        } finally {  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 423Exception in thread "main" MyExc3
- B. 4222222... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 42Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

12. Data la classe

```
1.  class X {  
2.      private Object o;  
3.      void f() {  
4.          o = new Object();  
5.          this.g(o);  
6.      }  
7.      void g(Object obj) {  
8.          obj = null;  
9.          obj = new Object();  
10.     }  
11.     public static void main(String[] a) {  
12.         new X().f();  
13.     }  
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

13. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

**14. Qual è l'output di questo codice?**

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

**15. Date le dichiarazioni:**

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
  - B. s = (C2) x;
  - C. c = (B2) s;
  - D. s = (C2) c;
  - E. Nessuno dei precedenti
- 

**16. Qual è l'output di questo codice?**

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 4**

1. Data la classe

```

1.   class X {
2.       private Object o;
3.       void f() {
4.           o = new Object();
5.           this.g(o);
6.       }
7.       void g(Object obj) {
8.           obj = null;
9.           obj = new Object();
10.      }
11.      public static void main(String[] a) {
12.          new X().f();
13.      }
14.  }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

2. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

3. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

4. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi static non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata final
- D. Un array vuoto può avere riferimento null
- E. L'operatore new non restituisce un riferimento all'oggetto appena creato

---

5. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}  
  
A. public A(String b) {}  
B. void A() {}  
C. private A(int i, String s) {}  
D. public Object A(int i, String s) {}  
E. Il costruttore di una classe non può essere sovraccaricato.
```

---

6. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args) {  
        A a = new B();  
        a.m();  
    }  
}
```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}
```

```
}  
  
public class B extends A {  
    B() {  
        super( 2 );  
    }  
    public static void main( String [] a) {  
        new B();  
    }  
}  
  
A. 0  
B. 1  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuna delle precedenti
```

---

8. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}  
  
public class B extends A {  
    public static void main( String [] a) {  
        new B();  
    }  
}  
  
A. 0  
B. 1  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuna delle precedenti
```

---

9. Quale output si ottiene invocando il metodo p?

```
class G {  
    private String s2 = new String("abcde");  
    private Integer i3 = new Integer(50);  
    void p() {  
        Integer i4;  
        i4 = i3;  
        m(new String("abcde"), new Boolean(true),  
           i4, i3);  
    }  
    void m(String s1, Boolean b2,  
          Integer i1, Integer i2) {  
        Boolean b1 = new Boolean(true);  
        if(s1 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```

}
if(b2 == b1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(i1 == i2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}

```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
- 

10. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

abstract class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}

```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 423Exception in thread "main" MyExc3
  - B. 4222222... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

---

E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Boolean [] c;  
Object [] [] s;  
Object [] w;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `c = (Boolean []) s;`
  - B. `s = w;`
  - C. `c = w;`
  - D. `s = (Object [] []) c;`
  - E. `w = (Object []) s;`
- 

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            q();  
        }  
        static void q() throws Exception {  
            try {  
                throw( new Exception() );  
            }  
            catch( MyExc2 d ) {  
                System.out.print(1);  
                throw( new MyExc3() );  
            }  
            catch( Exception h ) {  
                throw( new MyExc2() );  
            }  
            catch( MyExc3 w ) {  
            }  
            finally {  
                System.out.print(2);  
            }  
        }  
    }
```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
- 

15. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C2 extends C0 {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;  
C2 s;  
C0 x;  
c = new B2();  
x = new C2();  
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `c = (B2) x;`
  - B. `s = (C2) x;`
  - C. `c = (B2) s;`
  - D. `s = (C2) c;`
  - E. Nessuno dei precedenti
- 

16. Qual è l'output di questo codice?

```
public class A {  
    class B {  
        B() {  
            System.out.println("B");  
        }  
    }  
    A() {  
        System.out.println("A");  
    }  
    public static void main(String [] a) {  
        new A(). new B();  
    }  
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 5**

1. Data la classe

```

1.   class X {
2.       private Object o;
3.       void f() {
4.           o = new Object();
5.           this.g(o);
6.       }
7.       void g(Object obj) {
8.           obj = null;
9.           obj = new Object();
10.      }
11.      public static void main(String[] a) {
12.          new X().f();
13.      }
14.  }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

2. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        A a = new B();
```

- ```

        a.m();
    }
}
A. A.f
B. B.m
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti
```
- 

3. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}
```

```

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

#### 5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 423Exception in thread "main" MyExc3
  - B. 4222222... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

#### 6. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

#### 7. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 42
  - C. 341
  - D. 41
  - E. Nessuna delle precedenti
- 

#### 8. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;
- B. s = w;
- C. c = w;

D. `s = (Object [] []) c;`  
E. `w = (Object []) s;`

---

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
        finally {
            System.out.print(2);
        }
    }
}
```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}
public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}
A. 0
B. 1
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti
```

12. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        }
    }
}
```

```

    } else {
        System.out.print(0);
    }
}

```

- A. 000  
 B. 101  
 C. 100  
 D. 011  
 E. 001
- 

13. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)  
 B. public byte f(Object y)  
 C. protected int f(String z)  
 D. int f(String s) throws Exception  
 E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

abstract class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        I i = new B();
    }
}

```

```

        i.m();
    }
}
```

- A. A.f  
 B. B.m  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0  
 B. 1  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuna delle precedenti
- 

16. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
 B. Ai metodi static non si applica il *dynamic method dispatch*  
 C. Una classe non può essere dichiarata final  
 D. Un array vuoto può avere riferimento null  
 E. L'operatore new non restituisce un riferimento all'oggetto appena creato
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 6**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}
```

```
public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

3. Qual è il *sovraffunzione* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

4. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi *static* non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata final
- D. Un array vuoto può avere riferimento null
- E. L'operatore *new* non restituisce un riferimento all'oggetto appena creato

5. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 423Exception in thread "main" MyExc3
  - B. 4222222... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

#### 6. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;

- D. s = (C2) c;
  - E. Nessuno dei precedenti
- 

#### 7. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

#### 8. Date la classe

```

1.  class X {
2.      private Object o;
3.      void f() {
4.          o = new Object();
5.          this.g(o);
6.      }
7.      void g(Object obj) {
8.          obj = null;
9.          obj = new Object();
10.     }
11.    public static void main(String[] a) {
12.        new X().f();
13.    }
14. }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 8.
  - C. Al termine della linea 9.
  - D. Al termine della linea 12.
  - E. Non si può dire.
- 

#### 9. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;
- B. s = w;
- C. c = w;
- D. s = (Object [] []) c;
- E. w = (Object []) s;

---

10. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
abstract class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args) {  
        I i = new B();  
        i.m();  
    }  
}
```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}  
  
public class B extends A {  
    B() {  
        super( 2 );  
    }  
}
```

```
public static void main( String [] a) {  
    new B();  
}  
}  
  
A. 0  
B. 1  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuna delle precedenti
```

---

12. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class D1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            q();  
        }  
        static void q() throws Exception {  
            try {  
                throw( new Exception() );  
            }  
            catch( MyExc2 d ) {  
                System.out.print(1);  
                throw( new MyExc3() );  
            }  
            catch( Exception h ) {  
                throw( new MyExc2() );  
            }  
            catch( MyExc3 w ) {  
            }  
            finally {  
                System.out.print(2);  
            }  
        }  
    }  
}
```

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione

- C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
public class A {  
    class B {  
        B() {  
            System.out.println("B");  
        }  
    }  
    A() {  
        System.out.println("A");  
    }  
    public static void main(String [] a) {  
        new B();  
    }  
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```
class G {  
    private String s2 = new String("abcde");  
    private Integer i3 = new Integer(50);  
    void p() {  
        Integer i4;  
        i4 = i3;  
        m(new String("abcde"), new Boolean(true),  
          i4, i3);  
    }  
    void m(String s1, Boolean b2,  
          Integer i1, Integer i2) {  
        Boolean b1 = new Boolean(true);  
        if(s1 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        }  
    }  
}
```

```
} else {  
    System.out.print(0);  
}  
if(i1 == i2) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}  
}  
}  
}
```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
- 

16. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
}  
  
public static void main(String[] args){  
    A a = new B();  
    a.m();  
}
```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 7**

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
- 

2. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
}
```

```
A() {
    System.out.println("A");
}
public static void main(String [] a) {
    new B();
}
A. 0
B. 1
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti
```

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 423Exception in thread "main" MyExc3
- B. 4222222... (ciclo infinito)

- C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
        finally {
            System.out.print(2);
        }
    }
}
```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

5. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
}
```

```
}
```

- A. Errore a tempo di compilazione
  - B. 42
  - C. 341
  - D. 41
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
}

public static void main(String [] a) {
    new A(). new B();
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

8. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi `static` non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata `final`
- D. Un array vuoto può avere riferimento `null`
- E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

9. Date le dichiarazioni:

```
Boolean [] c;
Object [] [] s;
Object [] w;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

10. Qual è l'output di questo codice?

```
public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        A a = new B();
        a.m();
    }
}
```

- A. `A.f`
- B. `B.m`
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

11. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.   public static void main(String[] a) {
12.       new X().f();
13.   }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

12. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `c = (B2) x;`
- B. `s = (C2) x;`
- C. `c = (B2) s;`

- D. `s = (C2) c;`  
E. Nessuno dei precedenti
- 

13. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente `f` (overriding).

- A. `public void f(String x)`
  - B. `public byte f(Object y)`
  - C. `protected int f(String z)`
  - D. `int f(String s) throws Exception`
  - E. Nessuna delle precedenti.
- 

14. Qual è il *sovraffregamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. `public A(String b) {}`
  - B. `void A() {}`
  - C. `private A(int i, String s) {}`
  - D. `public Object A(int i, String s) {}`
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

15. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}
```

```
public class B extends A {
```

```
    B() {  
        super( 2 );  
    }  
    public static void main( String [] a) {  
        new B();  
    }  
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
abstract class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
}  
  
public static void main(String[] args) {  
    I i = new B();  
    i.m();  
}
```

- A. `A.f`
  - B. `B.m`
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 8**

1. Qual è l'output di questo codice?

```
public interface I {
    void m();
}

abstract class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}
```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
  - B. s = (C2) x;
  - C. c = (B2) s;
  - D. s = (C2) c;
  - E. Nessuno dei precedenti
- 

3. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

A. 0
B. 1
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti
```

---

4. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
```

```

        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0  
 B. 1  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuna delle precedenti
- 

5. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)  
 B. public byte f(Object y)  
 C. protected int f(String z)  
 D. int f(String s) throws Exception  
 E. Nessuna delle precedenti.
- 

6. Qual è il *sovraff장imento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
}
```

- A. public A(String b) {}  
 B. void A() {}  
 C. private A(int i, String s) {}  
 D. public Object A(int i, String s) {}  
 E. Il costruttore di una classe non può essere sovraccaricato.
- 

7. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,

```

```

        Integer i1, Integer i2) {
    Boolean b1 = new Boolean(true);
    if(s1 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 000  
 B. 101  
 C. 100  
 D. 011  
 E. 001
- 

8. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }
}

public static void main(String[] args) {
    A a = new B();
    a.m();
}

```

- A. A.f  
 B. B.m  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0  
 B. 1  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuna delle precedenti
- 

#### 10. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;  
 B. s = w;  
 C. c = w;  
 D. s = (Object [] []) c;  
 E. w = (Object []) s;
- 

#### 11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
    }
}

```

```

        catch( Exception h ) {
            throw( new MyExc2() );
        }
    catch( MyExc3 w ) {
    }
    finally {
        System.out.print(2);
    }
}

```

- A. 111... (ciclo infinito)  
 B. Errore a tempo di compilazione  
 C. Exception in thread "main" MyExc2  
 D. 2Exception in thread "main" MyExc2  
 E. Nessuna delle precedenti
- 

#### 12. Data la classe

```

1.   class X {
2.       private Object o;
3.       void f() {
4.           o = new Object();
5.           this.g(o);
6.       }
7.       void g(Object obj) {
8.           obj = null;
9.           obj = new Object();
10.      }
11.      public static void main(String[] a) {
12.          new X().f();
13.      }
14.  }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.  
 B. Al termine della linea 8.  
 C. Al termine della linea 9.  
 D. Al termine della linea 12.  
 E. Non si può dire.
- 

#### 13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

```

        finally {
            System.out.print(3);
        }
    }
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 a ) {
    }
    finally {
        throw( new MyExc3() );
    }
}
}

```

- A. 423Exception in thread "main" MyExc3  
B. 4222222... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 42Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi  
B. Ai metodi static non si applica il *dynamic method dispatch*  
C. Una classe non può essere dichiarata final  
D. Un array vuoto può avere riferimento null  
E. L'operatore new non restituisce un riferimento all'oggetto appena creato
- 

15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
}

```

```

        }
    }
    catch( MyExc2 h ) {
        System.out.print(2);
    }
}
static void q() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( Exception w ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione  
B. 42  
C. 341  
D. 41  
E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0  
B. 1  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 9**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
        finally {
            System.out.print(2);
        }
    }
}
```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}
```

```
}
```

```
public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

3. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a) {
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 8.
  - C. Al termine della linea 9.
  - D. Al termine della linea 12.
  - E. Non si può dire.
- 

4. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
- 

5. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
}
public static void main(String [] a) {
    new A(). new B();
}

```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

6. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

---

7. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
}
public static void main(String [] a) {
    new B();
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```
Boolean [] c;
Object [] [] s;
Object [] w;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;
  - B. s = w;
  - C. c = w;
  - D. s = (Object [] []) c;
  - E. w = (Object []) s;
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - B. Ai metodi **static** non si applica il *dynamic method dispatch*
  - C. Una classe non può essere dichiarata **final**
  - D. Un array vuoto può avere riferimento null
  - E. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
- 

11. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
- B. s = (C2) x;

- C. c = (B2) s;
  - D. s = (C2) c;
  - E. Nessuno dei precedenti
- 

12. Qual è l'output di questo codice?

```
public interface I {
    void m();
}

abstract class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}

A. A.f
B. B.m
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti
```

---

13. Qual è l'output di questo codice?

```
public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        A a = new B();
        a.m();
    }
}
```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

14. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 42
  - C. 341
  - D. 41
  - E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
```

```
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 423Exception in thread "main" MyExc3
  - B. 4222222... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

16. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 10**

1. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi **static** non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata **final**
- D. Un array vuoto può avere riferimento **null**
- E. L'operatore **new** non restituisce un riferimento all'oggetto appena creato

2. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

4. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

5. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

6. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

abstract class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}

```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

#### 7. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;
  - B. s = w;
  - C. c = w;
  - D. s = (Object [] []) c;
  - E. w = (Object []) s;
- 

#### 8. Data la classe

```

1.     class X {
2.         private Object o;
3.         void f() {
4.             o = new Object();
5.             this.g(o);
6.         }
7.         void g(Object obj) {
8.             obj = null;
9.             obj = new Object();

```

```

10.        }
11.        public static void main(String[] a) {
12.            new X().f();
13.        }
14.    }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 8.
  - C. Al termine della linea 9.
  - D. Al termine della linea 12.
  - E. Non si può dire.
- 

#### 9. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

```

```

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }
}

```

```

public static void main(String[] args) {
    A a = new B();
    a.m();
}

```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {

```

```

        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
        finally {
            System.out.print(2);
        }
    }
}

```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {

```

```

            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

13. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 42
  - C. 341
  - D. 41
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
- 

16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        } catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        } finally {
            System.out.print(3);
        }
        static void p() {
            try {
                System.out.print(4);
                throw( new Exception() );
            } catch( MyExc3 a ) {
            } finally {
                throw( new MyExc3() );
            }
        }
    }
}

```

- A. 423Exception in thread "main" MyExc3
  - B. 4222222... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 11**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
        finally {
            System.out.print(2);
        }
    }
}
```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

2. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
```

- A. public A(String b) {}
- B. void A() {}

- C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

3. Se nella classe C è dichiarato il metodo

int f(String s)

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

4. Data la classe

```
1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.    public static void main(String[] a) {
12.        new X().f();
13.    }
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 8.
  - C. Al termine della linea 9.
  - D. Al termine della linea 12.
  - E. Non si può dire.
-

5. Date le dichiarazioni:

```
Boolean [] c;  
Object [] [] s;  
Object [] w;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

---

6. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}  
  
public class B extends A {  
    B() {  
        super( 2 );  
    }  
    public static void main( String [] a) {  
        new B();  
    }  
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}  
  
public class B extends A {  
    public static void main( String [] a) {  
        new B();  
    }  
}
```

- A. 0

- B. 1

- C. Errore a tempo di compilazione

- D. Errore a tempo di esecuzione

- E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
abstract class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
}  
  
public static void main(String[] args){  
    I i = new B();  
    i.m();  
}
```

- A. A.f

- B. B.m

- C. Errore a tempo di compilazione

- D. Errore a tempo di esecuzione

- E. Nessuna delle precedenti

---

9. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi `static` non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata `final`
- D. Un array vuoto può avere riferimento `null`
- E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

---

10. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
  - B. s = (C2) x;
  - C. c = (B2) s;
  - D. s = (C2) c;
  - E. Nessuno dei precedenti
- 

11. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```
public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}
```

```
class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }
}

public static void main(String[] args) {
    A a = new B();
    a.m();
}
```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

13. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
- 

14. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

#### 15. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- }
  - }
  - }
  - A. 423Exception in thread "main" MyExc3
  - B. 4222222... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

#### 16. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 42
  - C. 341
  - D. 41
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 12

1. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi *static* non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata *final*
- D. Un array vuoto può avere riferimento *null*
- E. L'operatore *new* non restituisce un riferimento all'oggetto appena creato

2. Qual è il *sovvraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovraccaricato.

3. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C2 extends C0 {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;  
C2 s;  
C0 x;  
c = new B2();  
x = new C2();  
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```
public class A {  
    class B {  
        B() {  
            System.out.println("B");  
        }  
    }  
    A() {  
        System.out.println("A");  
    }  
    public static void main(String [] a) {  
        new A(). new B();  
    }  
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

5. Quale output si ottiene invocando il metodo p?

```
class G {  
    private String s2 = new String("abcde");  
    private Integer i3 = new Integer(50);  
    void p() {  
        Integer i4;  
        i4 = i3;  
        m(new String("abcde"), new Boolean(true),  
           i4, i3);  
    }  
    void m(String s1, Boolean b2,  
          Integer i1, Integer i2) {  
        Boolean b1 = new Boolean(true);  
        if(s1 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```

    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
- 

6. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

7. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;
- B. s = w;
- C. c = w;

```

D. s = (Object [] []) c;
E. w = (Object []) s;

```

---

8. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

abstract class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}

A. A.f
B. B.m
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

```

---

9. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        A a = new B();
        a.m();
    }
}

```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

10. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

11. Qual è l'output di questo codice?

```
public class A {  
    class B {  
        B() {  
            System.out.println("B");  
        }  
        A() {  
            System.out.println("A");  
        }  
    }  
    public static void main(String [] a) {  
        new B();  
    }  
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

12. Data la classe

```
1.     class X {  
2.         private Object o;  
3.         void f() {  
4.             o = new Object();  
5.             this.g(o);  
6.         }  
7.         void g(Object obj) {  
8.             obj = null;  
9.             obj = new Object();  
10.        }
```

```
11.         }  
12.         public static void main(String[] a) {  
13.             new X().f();  
14.         }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 8.
  - C. Al termine della linea 9.
  - D. Al termine della linea 12.
  - E. Non si può dire.
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            q();  
        }  
        static void q() throws Exception {  
            try {  
                throw( new Exception() );  
            }  
            catch( MyExc2 d ) {  
                System.out.print(1);  
                throw( new MyExc3() );  
            }  
            catch( Exception h ) {  
                throw( new MyExc2() );  
            }  
            catch( MyExc3 w ) {  
            }  
            finally {  
                System.out.print(2);  
            }  
        }  
    }
```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 42
  - C. 341
  - D. 41
  - E. Nessuna delle precedenti
- 

**15. Qual è l'output di questo codice?**

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

**16. Qual'è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 423Exception in thread "main" MyExc3
  - B. 4222222... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 13

1. Data la classe

```
1.     class X {  
2.         private Object o;  
3.         void f() {  
4.             o = new Object();  
5.             this.g(o);  
6.         }  
7.         void g(Object obj) {  
8.             obj = null;  
9.             obj = new Object();  
10.        }  
11.        public static void main(String[] a) {  
12.            new X().f();  
13.        }  
14.    }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

2. Qual è il *sovaccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
- B. void A() {}
- C. private A(int i, String s) {}
- D. public Object A(int i, String s) {}
- E. Il costruttore di una classe non può essere sovaccaricato.

3. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)
- B. public byte f(Object y)
- C. protected int f(String z)
- D. int f(String s) throws Exception
- E. Nessuna delle precedenti.

4. Qual è l'output di questo codice?

```
public class A {  
    class B {  
        B() {  
            System.out.println("B");  
        }  
    }  
    A() {  
        System.out.println("A");  
    }  
    public static void main(String [] a) {  
        new A(). new B();  
    }  
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

5. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- B. Ai metodi *static* non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata final
- D. Un array vuoto può avere riferimento null
- E. L'operatore *new* non restituisce un riferimento all'oggetto appena creato

6. Quale output si ottiene invocando il metodo p?

```
class G {  
    private String s2 = new String("abcde");  
    private Integer i3 = new Integer(50);  
    void p() {  
        Integer i4;  
        i4 = i3;  
        m(new String("abcde"), new Boolean(true),  
          i4, i3);  
    }  
    void m(String s1, Boolean b2,  
          Integer i1, Integer i2) {  
        Boolean b1 = new Boolean(true);  
        if(s1 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 000
- B. 101
- C. 100
- D. 011
- E. 001

7. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
}  
  
public static void main(String[] args){  
    A a = new B();
```

```
    a.m();  
}
```

- A. A.f
- B. B.m
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C2 extends C0 {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;  
C2 s;  
C0 x;  
c = new B2();  
x = new C2();  
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
- B. s = (C2) x;
- C. c = (B2) s;
- D. s = (C2) c;
- E. Nessuno dei precedenti

9. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}  
  
public class B extends A {  
    B() {  
        super( 2 );  
    }  
    public static void main( String [] a) {  
        new B();  
    }  
}  
  
A. 0  
B. 1  
C. Errore a tempo di compilazione
```

- D. Errore a tempo di esecuzione  
E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 423Exception in thread "main" MyExc3  
B. 4222222... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 42Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```
Boolean [] c;
Object [] [] s;
Object [] w;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;  
B. s = w;  
C. c = w;  
D. s = (Object [] []) c;  
E. w = (Object []) s;

---

12. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

A. 0
B. 1
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti
```

---

13. Qual è l'output di questo codice?

```
public interface I {
    void m();
}

abstract class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }
}

public static void main(String[] args) {
    I i = new B();
    i.m();
}
```

- A. A.f  
B. B.m  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuna delle precedenti
- 

14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 42  
 C. 341  
 D. 41  
 E. Nessuna delle precedenti
- 

**15. Qual'è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {

```

```

            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
        finally {
            System.out.print(2);
        }
    }
}

```

- A. 111... (ciclo infinito)  
 B. Errore a tempo di compilazione  
 C. Exception in thread "main" MyExc2  
 D. 2Exception in thread "main" MyExc2  
 E. Nessuna delle precedenti
- 

**16. Qual è l'output di questo codice?**

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0  
 B. 1  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 14**

1. Se nella classe C è dichiarato il metodo

- B. 101
  - C. 100
  - D. 011
  - E. 001
- 

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

2. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 000

3. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

A. 0
B. 1
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti
```

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
        }
    }
}
```

```

        throw( new MyExc3() );
    }
    catch( Exception h ) {
        throw( new MyExc2() );
    }
    catch( MyExc3 w ) {
    }
    finally {
        System.out.print(2);
    }
}

```

- A. 111... (ciclo infinito)  
B. Errore a tempo di compilazione  
C. Exception in thread "main" MyExc2  
D. 2Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0  
B. 1  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

abstract class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {

```

```

    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}

```

- A. A.f  
B. B.m  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuna delle precedenti
- 

#### 7. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0  
B. 1  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
}

```

```

static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 a ) {
    }
    finally {
        throw( new MyExc3() );
    }
}

```

- A. 423Exception in thread "main" MyExc3  
B. 4222222... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 42Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

#### 9. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;  
B. s = (C2) x;  
C. c = (B2) s;  
D. s = (C2) c;  
E. Nessuno dei precedenti
- 

#### 10. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
    }
}

```

```

        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 42  
C. 341  
D. 41  
E. Nessuna delle precedenti
- 

#### 11. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a ) {
        new B();
    }
}

```

- A. 0  
B. 1  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuna delle precedenti
- 

#### 12. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- 
- A. `c = (Boolean []) s;`
  - B. `s = w;`
  - C. `c = w;`
  - D. `s = (Object [] []) c;`
  - E. `w = (Object []) s;`
- 

13. Data la classe

```
1. class X {  
2.     private Object o;  
3.     void f() {  
4.         o = new Object();  
5.         this.g(o);  
6.     }  
7.     void g(Object obj) {  
8.         obj = null;  
9.         obj = new Object();  
10.    }  
11.    public static void main(String[] a) {  
12.        new X().f();  
13.    }  
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 8.
  - C. Al termine della linea 9.
  - D. Al termine della linea 12.
  - E. Non si può dire.
- 

14. Qual è il sovraccaricamento corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}  
  
A. public A(String b) {}  
B. void A() {}  
C. private A(int i, String s) {}  
D. public Object A(int i, String s) {}  
E. Il costruttore di una classe non può essere sovraccaricato.
```

---

15. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - B. Ai metodi `static` non si applica il *dynamic method dispatch*
  - C. Una classe non può essere dichiarata `final`
  - D. Un array vuoto può avere riferimento `null`
  - E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- 

16. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
}  
  
public static void main(String[] args) {  
    A a = new B();  
    a.m();  
}
```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 15**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
        finally {
            System.out.print(2);
        }
    }
}
```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}
```

```
}}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        A a = new B();
        a.m();
    }
}
```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

4. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

5. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C2 extends C0 {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;  
C2 s;  
C0 x;  
c = new B2();  
x = new C2();  
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
  - B. s = (C2) x;
  - C. c = (B2) s;
  - D. s = (C2) c;
  - E. Nessuno dei precedenti
- 

6. Qual è l'output di questo codice?

```
public class A {  
    class B {  
        B() {  
            System.out.println("B");  
        }  
    }  
    A() {  
        System.out.println("A");  
    }  
    public static void main(String [] a) {  
        new A(). new B();  
    }  
}
```

A. 0

B. 1

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuna delle precedenti

---

7. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - B. Ai metodi static non si applica il *dynamic method dispatch*
  - C. Una classe non può essere dichiarata final
  - D. Un array vuoto può avere riferimento null
  - E. L'operatore new non restituisce un riferimento all'oggetto appena creato
- 

8. Date le dichiarazioni:

```
Boolean [] c;  
Object [] [] s;  
Object [] w;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;
  - B. s = w;
  - C. c = w;
  - D. s = (Object [] []) c;
  - E. w = (Object []) s;
- 

9. Data la classe

```
1.  class X {  
2.      private Object o;  
3.      void f() {  
4.          o = new Object();  
5.          this.g(o);  
6.      }  
7.      void g(Object obj) {  
8.          obj = null;  
9.          obj = new Object();  
10.     }  
11.     public static void main(String[] a) {  
12.         new X().f();  
13.     }  
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.

- B. Al termine della linea 8.
  - C. Al termine della linea 9.
  - D. Al termine della linea 12.
  - E. Non si può dire.
- 

10. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        } catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 423Exception in thread "main" MyExc3
  - B. 4222222... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
```

```
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 42
  - C. 341
  - D. 41
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```
public interface I {
    void m();
}

abstract class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }
}

public static void main(String[] args) {
    I i = new B();
    i.m();
}
```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

13. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
- 

**14. Qual è l'output di questo codice?**

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

**15. Qual è l'output di questo codice?**

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

**16. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?**

```

public class A {
    A(String s) {}
}

A. public A(String b) {}
B. void A() {}
C. private A(int i, String s) {}
D. public Object A(int i, String s) {}
E. Il costruttore di una classe non può essere sovraccaricato.

```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 16**

1. Date le dichiarazioni:

```
class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `c = (B2) x;`
  - B. `s = (C2) x;`
  - C. `c = (B2) s;`
  - D. `s = (C2) c;`
  - E. Nessuno dei precedenti
- 

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
```

```
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
```

- A. 423Exception in thread "main" MyExc3
  - B. 4222222... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a ) {
        new B();
    }
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 42
  - C. 341
  - D. 41
  - E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

abstract class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}

```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

#### 6. Qual è il *sovraccaricamento* corretto del costruttore della seguente classe (overloading)?

- ```

public class A {
    A(String s) {}
}

```
- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

#### 7. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

#### 8. Data la classe

```

1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
}

```

```

9.           obj = new Object();
10.      }
11.      public static void main(String[] a) {
12.          new X().f();
13.      }
14.  }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 8.
  - C. Al termine della linea 9.
  - D. Al termine della linea 12.
  - E. Non si può dire.
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - B. Ai metodi *static* non si applica il *dynamic method dispatch*
  - C. Una classe non può essere dichiarata *final*
  - D. Un array vuoto può avere riferimento *null*
  - E. L'operatore *new* non restituisce un riferimento all'oggetto appena creato
- 

10. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

11. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

```

```

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        A a = new B();
        a.m();
    }
}

A. A.f
B. B.m
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

```

---

12. Quale output si ottiene invocando il metodo p?

```

class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
           i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
- 

13. Date le dichiarazioni:

```
Boolean [] c;
Object [] [] s;
Object [] w;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `c = (Boolean []) s;`
  - B. `s = w;`
  - C. `c = w;`
  - D. `s = (Object [] []) c;`
  - E. `w = (Object []) s;`
- 

14. Qual è l'output di questo codice?

```
class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
}
```

```
static void q() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc2 d ) {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( Exception h ) {
        throw( new MyExc2() );
    }
    catch( MyExc3 w ) {
    }
    finally {
        System.out.print(2);
    }
}
```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 17**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
        finally {
            System.out.print(2);
        }
    }
}
```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

2. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
- 

- C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

3. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
           i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
- 

4. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - B. Ai metodi `static` non si applica il *dynamic method dispatch*
  - C. Una classe non può essere dichiarata `final`
  - D. Un array vuoto può avere riferimento `null`
  - E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato
- 

5. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}  
  
public class B extends A {  
    B() {  
        super( 2 );  
    }  
    public static void main( String [] a) {  
        new B();  
    }  
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
public class A {  
    class B {  
        B() {  
            System.out.println("B");  
        }  
    }  
    A() {  
        System.out.println("A");  
    }  
    public static void main(String [] a) {  
        new B();  
    }  
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione

- D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
abstract class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
}  
  
public static void main(String[] args){  
    I i = new B();  
    i.m();  
}
```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C2 extends C0 {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;  
C2 s;  
C0 x;  
c = new B2();  
x = new C2();  
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `c = (B2) x;`
- B. `s = (C2) x;`
- C. `c = (B2) s;`
- D. `s = (C2) c;`
- E. Nessuno dei precedenti

---

9. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args) {  
        A a = new B();  
        a.m();  
    }  
}
```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}  
  
public class B extends A {  
    public static void main( String [] a) {  
        new B();  
    }  
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

11. Data la classe

```
1.     class X {  
2.         private Object o;  
3.         void f() {  
4.             o = new Object();  
5.             this.g(o);  
6.         }  
7.         void g(Object obj) {  
8.             obj = null;  
9.             obj = new Object();  
10.        }  
11.        public static void main(String[] a) {  
12.            new X().f();  
13.        }  
14.    }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 8.
  - C. Al termine della linea 9.
  - D. Al termine della linea 12.
  - E. Non si può dire.
- 

12. Qual è il *sovraffunzione* corretto del costruttore della seguente classe (overloading)?

```
public class A {  
    A(String s) {}  
}
```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

13. Date le dichiarazioni:

```
Boolean [] c;  
Object [] [] s;  
Object [] w;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;
  - B. s = w;
  - C. c = w;
  - D. s = (Object [] []) c;
  - E. w = (Object []) s;
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 42  
 C. 341  
 D. 41  
 E. Nessuna delle precedenti
- 

**15. Qual'è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

```

        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 423Exception in thread "main" MyExc3  
 B. 4222222... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 42Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

**16. Qual è l'output di questo codice?**

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0  
 B. 1  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 18

1. Date le dichiarazioni:

```
Boolean [] c;  
Object [] [] s;  
Object [] w;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `c = (Boolean []) s;`
- B. `s = w;`
- C. `c = w;`
- D. `s = (Object [] []) c;`
- E. `w = (Object []) s;`

2. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}  
  
public class B extends A {  
    public static void main( String [] a) {  
        new B();  
    }  
}  
  
A. 0  
B. 1  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuna delle precedenti
```

3. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

- B. Ai metodi `static` non si applica il *dynamic method dispatch*
- C. Una classe non può essere dichiarata `final`
- D. Un array vuoto può avere riferimento `null`
- E. L'operatore `new` non restituisce un riferimento all'oggetto appena creato

4. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
}  
  
public static void main(String[] args){  
    A a = new B();  
    a.m();  
}
```

- A. `A.f`
- B. `B.m`
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

5. Quale output si ottiene invocando il metodo `p`?

```
class G {  
    private String s2 = new String("abcde");  
    private Integer i3 = new Integer(50);  
    void p() {  
        Integer i4;
```

```

        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 000  
 B. 101  
 C. 100  
 D. 011  
 E. 001

6. Qual è il *sovvraccaricamento* corretto del costruttore della seguente classe (overloading)?

```
public class A {
    A(String s) {}
```

- A. public A(String b) {}  
 B. void A() {}  
 C. private A(int i, String s) {}  
 D. public Object A(int i, String s) {}  
 E. Il costruttore di una classe non può essere sovraccaricato.

7. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
```

```

        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

- A. 423Exception in thread "main" MyExc3  
 B. 4222222... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 42Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}
```

- A. 0  
 B. 1  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuna delle precedenti

9. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 42
  - C. 341
  - D. 41
  - E. Nessuna delle precedenti
- 

10. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
  - B. s = (C2) x;
  - C. c = (B2) s;
  - D. s = (C2) c;
  - E. Nessuno dei precedenti
- 

11. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

abstract class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}

```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

```

```

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0  
 B. 1  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuna delle precedenti
- 

#### 14. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
    finally {
        System.out.print(2);
    }
}

```

- }  
 A. 111... (ciclo infinito)  
 B. Errore a tempo di compilazione  
 C. Exception in thread "main" MyExc2  
 D. 2Exception in thread "main" MyExc2  
 E. Nessuna delle precedenti
- 

#### 15. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni *sovrascrive* correttamente f (overriding).

- A. public void f(String x)  
 B. public byte f(Object y)  
 C. protected int f(String z)  
 D. int f(String s) throws Exception  
 E. Nessuna delle precedenti.
- 

#### 16. Data la classe

```

1.   class X {
2.       private Object o;
3.       void f() {
4.           o = new Object();
5.           this.g(o);
6.       }
7.       void g(Object obj) {
8.           obj = null;
9.           obj = new Object();
10.      }
11.      public static void main(String[] a) {
12.          new X().f();
13.      }
14.  }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.  
 B. Al termine della linea 8.  
 C. Al termine della linea 9.  
 D. Al termine della linea 12.  
 E. Non si può dire.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 19**

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private String s2 = new String("abcde");
    private Integer i3 = new Integer(50);
    void p() {
        Integer i4;
        i4 = i3;
        m(new String("abcde"), new Boolean(true),
          i4, i3);
    }
    void m(String s1, Boolean b2,
           Integer i1, Integer i2) {
        Boolean b1 = new Boolean(true);
        if(s1 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
- 

2. Qual è l'output di questo codice?

```
public interface I {
    void m();
}

class A implements I {
    public void f() {
```

```
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        A a = new B();
        a.m();
    }
}

A. A.f
B. B.m
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti
```

---

3. Qual è l'output di questo codice?

```
public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new B();
    }
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
            System.out.print(1);
        }
        catch( MyExc2 h ) {
            System.out.print(2);
        }
    }
    static void q() throws Exception {
        try {
            throw( new MyExc2() );
        }
        catch( Exception w ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 42  
 C. 341  
 D. 41  
 E. Nessuna delle precedenti
- 

#### 5. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;  
 B. s = w;  
 C. c = w;  
 D. s = (Object [] []) c;  
 E. w = (Object []) s;
- 

#### 6. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)  
 B. public byte f(Object y)  
 C. protected int f(String z)  
 D. int f(String s) throws Exception  
 E. Nessuna delle precedenti.
- 

#### 7. Qual è l'output di questo codice?

```

public class A {
    class B {
        B() {
            System.out.println("B");
        }
    }
    A() {
        System.out.println("A");
    }
    public static void main(String [] a) {
        new A(). new B();
    }
}

```

- A. 0  
 B. 1  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuna delle precedenti
- 

#### 8. Qual'è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {

```

```

        throw( new MyExc3() );
    }
}

```

- A. 423Exception in thread "main" MyExc3
  - B. 4222222... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

class C0 extends Object {...}
class C2 extends C0 {...}
class B2 extends C0 {...}

```

e le inizializzazioni di variabile:

```

B2 c;
C2 s;
C0 x;
c = new B2();
x = new C2();
s = new C2();

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
  - B. s = (C2) x;
  - C. c = (B2) s;
  - D. s = (C2) c;
  - E. Nessuno dei precedenti
- 

10. Qual è il *sovvraccaricamento* corretto del costruttore della seguente classe (overloading)?

```

public class A {
    A(String s) {}
}

```

- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

11. Data la classe

```

1. class X {
2.     private Object o;
3.     void f() {
4.         o = new Object();
5.         this.g(o);
6.     }
7.     void g(Object obj) {
8.         obj = null;
9.         obj = new Object();
10.    }
11.   public static void main(String[] a) {
12.       new X().f();
13.   }
14. }

```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
  - B. Al termine della linea 8.
  - C. Al termine della linea 9.
  - D. Al termine della linea 12.
  - E. Non si può dire.
- 

12. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

public interface I {
    void m();
}

```

```

abstract class A implements I {
    public void f() {
}

```

```

        System.out.println("A.f");
    }

}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        I i = new B();
        i.m();
    }
}

A. A.f
B. B.m
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

```

---

**14. Qual'è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
        finally {
            System.out.print(2);
        }
    }
}

```

- A. 111... (ciclo infinito)
  - B. Errore a tempo di compilazione
  - C. Exception in thread "main" MyExc2
  - D. 2Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

**15. Qual è l'output di questo codice?**

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a ) {
        new B();
    }
}

A. 0
B. 1
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

```

---

**16. Dire quale delle seguenti affermazioni è vera:**

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - B. Ai metodi **static** non si applica il *dynamic method dispatch*
  - C. Una classe non può essere dichiarata final
  - D. Un array vuoto può avere riferimento null
  - E. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 20**

1. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            q();
        }
    }
    static void q() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc2 d ) {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( Exception h ) {
            throw( new MyExc2() );
        }
        catch( MyExc3 w ) {
        }
        finally {
            System.out.print(2);
        }
    }
}
```

- A. 111... (ciclo infinito)
- B. Errore a tempo di compilazione
- C. Exception in thread "main" MyExc2
- D. 2Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
```

```
        q();
        System.out.print(1);
    }
    catch( MyExc2 h ) {
        System.out.print(2);
    }
}
static void q() throws Exception {
    try {
        throw( new MyExc2() );
    }
    catch( Exception w ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
    }
}
```

- A. Errore a tempo di compilazione
- B. 42
- C. 341
- D. 41
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
public interface I {
    void m();
}

class A implements I {
    public void f() {
        System.out.println("A.f");
    }
}

class B extends A {
    public void m() {
        System.out.println("B.m");
    }
    public void f() { m(); }

    public static void main(String[] args) {
        A a = new B();
```

```

    a.m();
}
A. A.f
B. B.m
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuna delle precedenti

```

---

4. Qual è l'output di questo codice?

```

class A {
    A() {
        System.out.println(0);
    }
    A( int i ) {
        System.out.println(i);
    }
}

public class B extends A {
    B() {
        super( 2 );
    }
    public static void main( String [] a) {
        new B();
    }
}

```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
  - B. Ai metodi **static** non si applica il *dynamic method dispatch*
  - C. Una classe non può essere dichiarata **final**
  - D. Un array vuoto può avere riferimento null
  - E. L'operatore **new** non restituisce un riferimento all'oggetto appena creato
- 

6. Se nella classe C è dichiarato il metodo

```
int f(String s)
```

indicare quale delle seguenti ulteriori dichiarazioni sovrascrive correttamente f (overriding).

- A. public void f(String x)
  - B. public byte f(Object y)
  - C. protected int f(String z)
  - D. int f(String s) throws Exception
  - E. Nessuna delle precedenti.
- 

7. Date le dichiarazioni:

```

Boolean [] c;
Object [] [] s;
Object [] w;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. c = (Boolean []) s;
  - B. s = w;
  - C. c = w;
  - D. s = (Object [] []) c;
  - E. w = (Object []) s;
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class D1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 r ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 a ) {
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

- A. 423Exception in thread "main" MyExc3
- B. 4222222... (ciclo infinito)

- C. Errore a tempo di compilazione
  - D. 42Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```
public interface I {  
    void m();  
}  
  
abstract class A implements I {  
    public void f() {  
        System.out.println("A.f");  
    }  
}  
  
class B extends A {  
    public void m() {  
        System.out.println("B.m");  
    }  
    public void f() { m(); }  
  
    public static void main(String[] args) {  
        I i = new B();  
        i.m();  
    }  
}
```

- A. A.f
  - B. B.m
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
public class A {  
    class B {  
        B() {  
            System.out.println("B");  
        }  
    }  
    A() {  
        System.out.println("A");  
    }  
    public static void main(String [] a) {  
        new A(). new B();  
    }  
}
```

- A. 0
  - B. 1
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
- 

- E. Nessuna delle precedenti
- 

11. Qual è il *sovraffunzionamento* corretto del costruttore della seguente classe (overloading)?

- ```
public class A {  
    A(String s) {}  
}
```
- A. public A(String b) {}
  - B. void A() {}
  - C. private A(int i, String s) {}
  - D. public Object A(int i, String s) {}
  - E. Il costruttore di una classe non può essere sovraccaricato.
- 

12. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C2 extends C0 {...}  
class B2 extends C0 {...}
```

e le inizializzazioni di variabile:

```
B2 c;  
C2 s;  
C0 x;  
c = new B2();  
x = new C2();  
s = new C2();
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. c = (B2) x;
  - B. s = (C2) x;
  - C. c = (B2) s;
  - D. s = (C2) c;
  - E. Nessuno dei precedenti
- 

13. Qual è l'output di questo codice?

```
class A {  
    A() {  
        System.out.println(0);  
    }  
    A( int i ) {  
        System.out.println(i);  
    }  
}  
  
public class B extends A {  
    public static void main( String [] a) {  
        new B();  
    }  
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

---

#### 14. Data la classe

```
1. class X {  
2.     private Object o;  
3.     void f() {  
4.         o = new Object();  
5.         this.g(o);  
6.     }  
7.     void g(Object obj) {  
8.         obj = null;  
9.         obj = new Object();  
10.    }  
11.    public static void main(String[] a){  
12.        new X().f();  
13.    }  
14. }
```

Quando l'oggetto creato alla linea 4 diventa eleggibile per la garbage collection?

- A. Al termine della linea 5.
- B. Al termine della linea 8.
- C. Al termine della linea 9.
- D. Al termine della linea 12.
- E. Non si può dire.

---

#### 15. Qual è l'output di questo codice?

```
public class A {  
    class B {  
        B() {  
            System.out.println("B");  
        }  
    }  
    A() {  
        System.out.println("A");  
    }  
    public static void main(String [] a) {  
        new B();  
    }  
}
```

- A. 0
- B. 1
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuna delle precedenti

---

#### 16. Quale output si ottiene invocando il metodo p?

```
class G {  
    private String s2 = new String("abcde");  
    private Integer i3 = new Integer(50);  
    void p() {  
        Integer i4;  
        i4 = i3;  
        m(new String("abcde"), new Boolean(true),  
           i4, i3);  
    }  
    void m(String s1, Boolean b2,  
           Integer i1, Integer i2) {  
        Boolean b1 = new Boolean(true);  
        if(s1 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 000
  - B. 101
  - C. 100
  - D. 011
  - E. 001
-

**Prova n. 1**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 2**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 3**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 4**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 5**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 6**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 7**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 8**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 9**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 10**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 11**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 12**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 13**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 15**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 16**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

12 gennaio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

## Risultato prova n. 1:

1			■				
2				■			
3						■	■
4						■	■
5					■	■	
6						■	■
7		■	■				
8				■	■		
9						■	■
10			■	■			
11	■	■					
12				■	■		
13				■	■		
14				■	■		
15			■	■			
16				■	■		

Risultato prova n. 2:

1							1
2							2
3			2				3
4			2				4
5			2				5
6	2						6
7						2	7
8			2				8
9		2					9
10			2				10
11			2				11
12				2			12
13			2				13
14				2			14
15		2					15
16					2		16

Risultato prova n. 3:

1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Risultato prova n. 4:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 5:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 6:

1							1
2							2
3							3
4							4
5							5
6							6
7							7
8							8
9							9
10							10
11							11
12							12
13							13
14							14
15							15
16							16

Risultato prova n. 7:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 8:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 9:

1			■				
2	■						
3		■					
4				■	■		
5				■	■		
6		■					
7						■	
8					■	■	
9				■	■		
10						■	
11		■					
12						■	
13				■	■		
14				■	■		
15						■	
16			■				

Risultato prova n. 10:

1			■					
2				■	■			
3				■	■			
4						■		
5							■	■
6							■	■
7	■	■						
8			■	■				
9			■	■				
10			■	■				
11					■	■		
12					■	■		
13							■	■
14							■	■
15				■	■			
16				■	■			

Risultato prova n. 11:

1			■				
2				■	■		
3			■				
4						■	■
5						■	■
6	■	■					
7						■	■
8		■	■				
9				■	■		
10				■	■		
11				■	■		
12					■	■	
13		■	■				
14				■	■		
15						■	■
16				■	■		

Risultato prova n. 12:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 13:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 14:

1			■				
2				■			
3						■	
4				■			
5		■					
6						■	
7		■					
8						■	
9					■		
10				■			
11				■			
12		■					
13						■	
14	■						
15			■				
16			■				

Risultato prova n. 15:

1			■				
2				■			
3						■	
4				■			
5		■					
6			■				
7						■	
8					■		
9		■					
10				■			
11			■				
12						■	
13						■	
14	■						
15		■					
16			■				

Risultato prova n. 16:

1			■					
2				■				
3							■	
4			■					
5							■	
6				■				
7			■					
8			■					
9				■				
10	■							
11					■			
12				■				
13						■		
14				■				
15				■				
16						■		

Risultato prova n. 17:

## Risultato prova n. 18:

## Risultato prova n. 19:

1			■				
2				■	■		
3				■	■		
4						■	
5		■	■				
6				■	■		
7						■	
8				■	■		
9			■	■			
10						■	
11				■	■		
12			■	■			
13	■	■					
14				■	■		
15				■	■		
16						■	

Risultato prova n. 20:



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 1**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
```

```
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. 42

B. 41

C. 4

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

4. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

5. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

6. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {
```

```
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

A. 1

B. 2.0

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuno dei precedenti

---

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

8. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

9. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println( _QUI_ );  
        }  
    }  
}
```

A. x

B. A.this.x

C. A.x

D. super.x

E. Nessuna delle precedenti

10. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

A. 1

B. 2

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuno dei precedenti

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
            System.out.print(1);  
        }  
        catch( MyExc3 h ) {  
            throw( new MyExc2() );  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 k ) {  
            throw( new MyExc2() );  
        }  
        catch( Exception e ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
    }  
}
```

}

}

A. 23431

B. Errore a tempo di compilazione

C. 234Exception in thread "main" java.lang.Exception

D. 24Exception in thread "main" java.lang.Exception

E. Nessuna delle precedenti

12. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

A. x

B. A.this.x

C. A.x

D. super.x

E. Nessuna delle precedenti

13. Quale output si ottiene invocando il metodo p?

```
class G {  
    private Boolean b1 = new Boolean(false);  
    private String s2 = new String("");  
    private String s3;  
    void p() {  
        m(b1, new String(""));  
    }  
    void m(Boolean b2, String s1) {  
        String s4;  
        s3 = "";  
        s4 = "";  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```

        System.out.print(0);
    }
    if(s4 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 110  
 B. 011  
 C. 001  
**D. 101**  
 E. 010
- 

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1**  
 B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

15. Date la classe

---

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che **n.info=""**, dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();  
 B. int l = n.getInfo().length();  
**C. n=new Nodo(new Integer(0));**  
 D. int v = n.getInfo().intValue();  
 E. Nessuno dei precedenti
- 

16. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
**B. 2**  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 2**

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101**
- E. 010

2. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {

```

```
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1**
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione**

- C. 234Exception in thread "main" java.lang.Exception  
D. 24Exception in thread "main" java.lang.Exception  
E. Nessuna delle precedenti
- 

4. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

- A. 1  
B. 2.0  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuno dei precedenti
- 

5. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;  
B. f = (Boolean []) u;  
C. u = (Exception []) e;  
D. u = (Exception []) f;  
E. Nessuno dei precedenti
- 

6. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null

- B. Un oggetto può essere inizializzato dopo la sua dichiarazione  
C. Un array possiede dei membri  
D. Ai metodi static non si applica il *dynamic method dispatch*  
E. Una classe interna può essere dichiarata private
- 

7. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");  
B. Nodo<String> n = new Nodo<String>("");  
C. new Nodo<Boolean>("");  
D. new Nodo<Boolean>(true).getInfo().length()  
E. Nessuno dei precedenti
- 

8. Qual è il risultato di questo codice:

```
class A {  
    static protected int x = 1;  
    public static int get() { return x; }  
}  
public class B extends A {  
    public static int get() { return x+1; }  
    public static void main(String[] s) {  
        A a = new B();  
        B b = (B) a;  
        System.out.println( a.get() + b.get() );  
    }  
}
```

- A. 12  
B. 22  
C. 3  
D. 4  
E. Nessuno dei precedenti
- 

9. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();  
B. int l = n.getInfo().length();  
**C. n=new Nodo(new Integer(0));**  
D. int v = n.getInfo().intValue();  
E. Nessuno dei precedenti
- 

10. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;  
B. e = (Boolean) m;  
C. e = v;  
D. m = (Exception) e;  
**E. v = e;**
- 

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

- A. x  
B. A.this.x  
C. A.x  
**D. super.x**  
E. Nessuna delle precedenti
- 

12. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1**  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class D1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            m();  
        }  
        catch( Exception x ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 y ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 u ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 42  
B. 41  
C. 4  
**D. Errore a tempo di compilazione**  
E. Nessuna delle precedenti
- 

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
**B. 2**  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
    }
}

```

```

        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452**  
 B. 352Exception in thread "main" MyExc2  
 C. 35  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

16. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x  
 B. A.this.x  
**C. A.x**  
 D. super.x  
 E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 3**

1. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

2. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

3. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

4. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
}
public static void main(String a[]) {
    B obj = new B();
    obj.x = 1;
    obj.set_x(2);
    obj.my_print();
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

5. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println(_QUI_ );  
        }  
    }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

6. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

7. Quale output si ottiene invocando il metodo p?

```
class G {  
    private Boolean b1 = new Boolean(false);  
    private String s2 = new String("");  
    private String s3;  
    void p() {  
        m(b1, new String(""));  
    }  
    void m(Boolean b2, String s1) {  
        String s4;  
        s3 = "";  
        s4 = "";  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {
```

```
            System.out.print(0);  
        }  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s3) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}  
  


- A. 110
- B. 011
- C. 001
- D. 101
- E. 010



---



8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?



```
public class C {  
    String x = "abc";  
    Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```



- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti



---


```

9. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;

E. v = e;

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. 42

B. 41

C. 4

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
    finally {
        System.out.print(2);
    }
}
```

```
}
```

- ```
static void n() throws Exception {
    try {
        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
```
- A. 3452  
B. 352Exception in thread "main" MyExc2  
C. 35  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

---

12. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null  
B. Un oggetto può essere inizializzato dopo la sua dichiarazione  
C. Un array possiede dei membri  
D. Ai metodi static non si applica il *dynamic method dispatch*  
E. Una classe interna può essere dichiarata private

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
    }
}
```

```

        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431  
B. Errore a tempo di compilazione  
C. 234Exception in thread "main" java.lang.Exception  
D. 24Exception in thread "main" java.lang.Exception  
E. Nessuna delle precedenti
- 

14. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
}
public static void main(String a[]) {
    new B();
}

```

- A. 1  
B. 2.0  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuno dei precedenti

---

15. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;  
B. f = (Boolean []) u;  
C. u = (Exception []) e;  
D. u = (Exception []) f;  
E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 4**

1. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

2. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;
    C(Integer y) { this.y = y; }
```

```
public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

4. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

5. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
```

```

public static void main(String[] s) {
    A a = new B();
    B b = (B) a;
    System.out.println( a.get() + b.get() );
}
}

```

- A. 12  
 B. 22  
 C. 3  
 D. 4  
 E. Nessuno dei precedenti
- 

## 6. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");  
 B. Nodo<String> n = new Nodo<String>("");  
 C. new Nodo<Boolean>("");  
 D. new Nodo<Boolean>(true).getInfo().length();  
 E. Nessuno dei precedenti
- 

## 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
    }
}

```

```

    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}
}

A. 23431  

B. Errore a tempo di compilazione  

C. 234Exception in thread "main" java.lang.Exception  

D. 24Exception in thread "main" java.lang.Exception  

E. Nessuna delle precedenti
```

---

## 8. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
}
public static void main(String a[]) {
    new B();
}

```

- A. 1  
 B. 2.0  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuno dei precedenti
- 

## 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
    }
}

```

```

        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42  
 B. 41  
 C. 4  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

10. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x  
 B. A.this.x  
 C. A.x  
 D. super.x  
 E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        }
    }
}

```

```

        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
 B. 011  
 C. 001  
 D. 101  
 E. 010
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

13. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

14. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

---

15. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

---

16. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 5**

1. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

3. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

4. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

5. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

7. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
```

```
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

8. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

9. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3

D. 4

E. Nessuno dei precedenti

- 
10. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println(_QUI_ );  
        }  
    }  
}
```

- A. x  
B. A.this.x  
C. A.x  
D. super.x  
E. Nessuna delle precedenti

- 
11. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;  
B. f = (Boolean []) u;  
C. u = (Exception []) e;  
D. u = (Exception []) f;  
E. Nessuno dei precedenti

- 
12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class D1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            m();  
        }  
        catch( Exception x ) {  
            System.out.print(1);  
        }  
    }  
}
```

```
}  
catch( MyExc2 y ) {  
    System.out.print(2);  
}  
catch( MyExc3 u ) {  
    System.out.print(3);  
}  
}  
}  
static void m() throws Exception {  
    try {  
        throw( new MyExc2() );  
    }  
    finally {  
        System.out.print(4);  
    }  
}  
}
```

- A. 42  
B. 41  
C. 4  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

- 
13. Quale output si ottiene invocando il metodo p?

```
class G {  
    private Boolean b1 = new Boolean(false);  
    private String s2 = new String("");  
    private String s3;  
    void p() {  
        m(b1, new String(""));  
    }  
    void m(Boolean b2, String s1) {  
        String s4;  
        s3 = "";  
        s4 = "";  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s3) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110  
B. 011  
C. 001  
D. 101

## 14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

## 15. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

## 16. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 6**

1. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println(_QUI_);
        }
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
```

```
public static void main(String [] argv)
throws Exception {
    try {
        n();
    }
    catch( MyExc3 j ) {
        System.out.print(1);
    }
    finally {
        System.out.print(2);
    }
}
static void n() throws Exception {
    try {
        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
```

```

float x;
B() {
    x = 2;
    System.out.print(x);
}
public static void main(String a[]) {
    new B();
}

```

- A. 1  
 B. 2.0  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuno dei precedenti
- 

5. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12  
 B. 22  
 C. 3  
 D. 4  
 E. Nessuno dei precedenti
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {

```

```

            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}
A. 42  

B. 41  

C. 4  

D. Errore a tempo di compilazione  

E. Nessuna delle precedenti
```

---

7. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
 B. 011  
 C. 001  
 D. 101  
 E. 010
- 

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

10. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi **static** non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
}

```

```

public void my_print() {
    System.out.println( _QUI_ );
}
public static void main(String a[]) {
    B obj = new B();
    obj.x = 1;
    obj.set_x(2);
    obj.my_print();
}

```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

12. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

13. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
-

#### 14. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

---

#### 15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
            System.out.print(1);  
        }  
        catch( MyExc3 h ) {  
            throw( new MyExc2() );  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 k ) {  
            throw( new MyExc2() );  
        }  
    }  
}
```

```
    catch( Exception e ) {  
        System.out.print(3);  
    }  
    finally {  
        System.out.print(4);  
        throw( new Exception() );  
    }  
}  
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

#### 16. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 7**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

A. x
B. A.this.x
C. A.x
D. super.x
E. Nessuna delle precedenti
```

---

4. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
}
```

```
    }
}
static void n() throws Exception {
    try {
        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

8. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
}
```

```

}
public static void main(String a[]) {
    B obj = new B();
    obj.x = 1;
    obj.set_x(2);
    obj.my_print();
}
}

```

- A. x  
B. A.this.x  
C. A.x  
D. super.x  
E. Nessuna delle precedenti
- 

9. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1  
B. 2  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuno dei precedenti
- 

10. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

if(s2 == s1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
A. 110
B. 011
C. 001
D. 101
E. 010

```

---

11. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;  
B. f = (Boolean []) u;  
C. u = (Exception []) e;  
D. u = (Exception []) f;  
E. Nessuno dei precedenti
- 

12. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;
    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4

E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

15. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
```

```
float x;
B() {
    x = 2;
    System.out.print(x);
}
public static void main(String a[]) {
    new B();
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

16. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 8**

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

2. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
```

```
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println(_QUI_);
        }
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
            try {

```

```

        m();
    }
    catch( Exception x ) {
        System.out.print(1);
    }
    catch( MyExc2 y ) {
        System.out.print(2);
    }
    catch( MyExc3 u ) {
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}

```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione

- C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

9. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

10. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = v;`
  - B. `e = (Boolean) m;`
  - C. `e = v;`
  - D. `m = (Exception) e;`
  - E. `v = e;`
- 

11. Qual è il risultato di questo codice:

```
class A {  
    static protected int x = 1;  
    public static int get() { return x; }  
}  
public class B extends A {  
    public static int get() { return x+1; }  
    public static void main(String[] s) {  
        A a = new B();  
        B b = (B) a;  
        System.out.println( a.get() + b.get() );  
    }  
}  
  
A. 12  
B. 22  
C. 3  
D. 4  
E. Nessuno dei precedenti
```

---

12. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
  - B. `Nodo<String> n = new Nodo<String>("");`
  - C. `new Nodo<Boolean>("")`
  - D. `new Nodo<Boolean>(true).getInfo().length()`
  - E. Nessuno dei precedenti
- 

13. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

e una variabile n di tipo Nodo tale che n.info=""; dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

14. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

15. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
```

```
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

A. 1
B. 2
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuno dei precedenti
```

---

16. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 9**

1. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

- D. super.x  
E. Nessuna delle precedenti
- 

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

2. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x  
B. A.this.x  
C. A.x
- 

3. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110  
B. 011  
C. 001  
D. 101  
E. 010
- 

4. Date la classe

```
class Nodo {
    private Object info;
```

```

Nodo next;
Nodo(Object i) { info=i; }
public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
}
public static void main(String a[])
{
    new B();
}

```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione

E. Nessuno dei precedenti

---

8. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

9. Qual è il risultato di questo codice:

```
class A {  
    static protected int x = 1;  
    public static int get() { return x; }  
}  
  
public class B extends A {  
    public static int get() { return x+1; }  
    public static void main(String[] s) {  
        A a = new B();  
        B b = (B) a;  
        System.out.println( a.get() + b.get() );  
    }  
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

10. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

11. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}  
  
A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
```

---

12. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

13. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

---

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

A. x
B. A.this.x
C. A.x
D. super.x
E. Nessuna delle precedenti

```

---

16. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

A. 1
B. 2
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuno dei precedenti

```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 10**

1. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

2. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione

- D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

3. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
    }
}
```

```

        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431  
 B. Errore a tempo di compilazione  
 C. 234Exception in thread "main" java.lang.Exception  
 D. 24Exception in thread "main" java.lang.Exception  
 E. Nessuna delle precedenti
- 

5. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null  
 B. Un oggetto può essere inizializzato dopo la sua dichiarazione  
 C. Un array possiede dei membri  
 D. Ai metodi static non si applica il *dynamic method dispatch*  
 E. Una classe interna può essere dichiarata private
- 

6. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x  
 B. A.this.x  
 C. A.x  
 D. super.x  
 E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42  
 B. 41  
 C. 4  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;  
 B. e = (Boolean) m;  
 C. e = v;  
 D. m = (Exception) e;  
 E. v = e;
- 

9. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

10. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println(_QUI_);
        }
    }
}

```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

#### 11. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("") ;
  - B. Nodo<String> n = new Nodo<String>("") ;
  - C. new Nodo<Boolean>("") ;
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

12. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

13. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2

C. 3

D. 4

E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

A. 3452

B. 352Exception in thread "main" MyExc2

C. 35

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

15. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

16. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

A. 12

B. 22

C. 3

D. 4

E. Nessuno dei precedenti

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 11**

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

2. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

3. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("") ;
  - B. Nodo<String> n = new Nodo<String>("") ;
  - C. new Nodo<Boolean>("") ;
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

4. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

5. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
```

```
protected int x;
public static void main(String a[]) {
    B obj = new B();
    obj.x = 1;
    obj.set_x(2);
    System.out.println( obj.x );
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

8. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
```

```

        throw( new MyExc2() );
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}

```

- A. 23431  
B. Errore a tempo di compilazione  
C. 234Exception in thread "main" java.lang.Exception  
D. 24Exception in thread "main" java.lang.Exception  
E. Nessuna delle precedenti
- 

10. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();  
B. int l = n.getInfo().length();  
C. n=new Nodo(new Integer(0));  
D. int v = n.getInfo().intValue();  
E. Nessuno dei precedenti
- 

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x  
B. A.this.x  
C. A.x  
D. super.x  
E. Nessuna delle precedenti
- 

12. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;
    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

13. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1  
B. 2.0  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuno dei precedenti
-

14. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

15. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

---

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class C1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            n();  
        }  
        catch( MyExc3 j ) {  
            System.out.print(1);  
        }  
        finally {  
            System.out.print(2);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            System.out.print(3);  
            if (true) throw( new MyExc2() );  
            else throw( new MyExc3() );  
        }  
        catch( MyExc1 g ) {  
            System.out.print(4);  
        }  
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 12

1. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

2. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

A. x
B. A.this.x
C. A.x
D. super.x
E. Nessuna delle precedenti
```

4. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

5. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12  
 B. 22  
 C. 3  
 D. 4  
 E. Nessuno dei precedenti
- 

6. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1  
 B. 2  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuno dei precedenti
- 

7. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
}

```

```

    }
    public static void main(String a[]) {
        new B();
    }
}

A. 1
B. 2.0
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuno dei precedenti

```

---

8. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null  
 B. Un oggetto può essere inizializzato dopo la sua dichiarazione  
 C. Un array possiede dei membri  
 D. Ai metodi static non si applica il *dynamic method dispatch*  
 E. Una classe interna può essere dichiarata private
- 

9. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

10. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");

- B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
}

```

```

        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {

```

```
        System.out.print(5);
    }
}
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

14. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011

C. 001

D. 101

E. 010

---

15. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

---

16. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 13**

1. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

2. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

3. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

4. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

5. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

#### 8. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

#### 9. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x

- B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

11. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;
    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
```

```
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

12. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("") ;
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

13. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

14. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1  
 B. 2  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuno dei precedenti
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
}

```

```

    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42  
 B. 41  
 C. 4  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

16. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12  
 B. 22  
 C. 3  
 D. 4  
 E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 14**

1. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

2. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;
    C(Integer y) { this.y = y; }
    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
A. 1
B. 2
C. 3
D. 4
E. Nessuna delle precedenti
```

---

4. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

5. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info=""; dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

#### 6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
            try {

```

```
                n();
            }
            catch( MyExc3 j ) {
                System.out.print(1);
            }
            finally {
                System.out.print(2);
            }
        }
        static void n() throws Exception {
            try {
                System.out.print(3);
                if (true) throw( new MyExc2() );
                else throw( new MyExc3() );
            }
            catch( MyExc1 g ) {
                System.out.print(4);
            }
            catch( MyExc3 e ) {
                throw( new MyExc1() );
            }
            finally {
                System.out.print(5);
            }
        }
    }
```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
    }
}
```

```

        }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}

```

- A. 23431  
 B. Errore a tempo di compilazione  
 C. 234Exception in thread "main" java.lang.Exception  
 D. 24Exception in thread "main" java.lang.Exception  
 E. Nessuna delle precedenti
- 

9. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

10. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12  
 B. 22  
 C. 3

D. 4

E. Nessuno dei precedenti

---

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

A. x
B. A.this.x
C. A.x
D. super.x
E. Nessuna delle precedenti

```

---

12. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;  
 B. e = (Boolean) m;  
 C. e = v;  
 D. m = (Exception) e;  
 E. v = e;
- 

13. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null  
 B. Un oggetto può essere inizializzato dopo la sua dichiarazione  
 C. Un array possiede dei membri  
 D. Ai metodi static non si applica il *dynamic method dispatch*  
 E. Una classe interna può essere dichiarata private
- 

14. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1  
 B. 2.0  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuno dei precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {

```

```

            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

A. 110
B. 011
C. 001
D. 101
E. 010

```

---

16. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println(_QUI_);
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

A. x
B. A.this.x
C. A.x
D. super.x
E. Nessuna delle precedenti

```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 15**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

2. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
```

```
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}  
  
A. x  
B. A.this.x  
C. A.x  
D. super.x  
E. Nessuna delle precedenti
```

---

3. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

4. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;

- D. u = (Exception []) f;  
E. Nessuno dei precedenti
- 

5. Qual è il risultato di questo codice:

```
class A {  
    static protected int x = 1;  
    public static int get() { return x; }  
}  
public class B extends A {  
    public static int get() { return x+1; }  
    public static void main(String[] s) {  
        A a = new B();  
        B b = (B) a;  
        System.out.println( a.get() + b.get() );  
    }  
}
```

- A. 12  
B. 22  
C. 3  
D. 4  
E. Nessuno dei precedenti
- 

6. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class D1 {  
    public static void main(String [] argv)
```

```
throws Exception {  
    try {  
        m();  
    }  
    catch( Exception x ) {  
        System.out.print(1);  
    }  
    catch( MyExc2 y ) {  
        System.out.print(2);  
    }  
    catch( MyExc3 u ) {  
        System.out.print(3);  
    }  
}  
static void m() throws Exception {  
    try {  
        throw( new MyExc2() );  
    }  
    finally {  
        System.out.print(4);  
    }  
}
```

- A. 42  
B. 41  
C. 4  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

8. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

- A. 1  
B. 2.0  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuno dei precedenti
- 

9. Dato il template

---

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

12. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="",  
dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

13. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

---

14. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println( _QUI_ );  
        }  
    }  
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

15. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}
```

```
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

---

16. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 16

1. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

2. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println( _QUI_ );  
        }  
    }  
}  
  
A. x  
B. A.this.x  
C. A.x  
D. super.x  
E. Nessuna delle precedenti
```

3. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;

E. v = e;

4. Qual è il risultato di questo codice:

```
class A {  
    static protected int x = 1;  
    public static int get() { return x; }  
}  
public class B extends A {  
    public static int get() { return x+1; }  
    public static void main(String[] s) {  
        A a = new B();  
        B b = (B) a;  
        System.out.println( a.get() + b.get() );  
    }  
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

5. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
```

```
        System.out.print(3);
    }
}
static void m() throws Exception {
    try {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(4);
    }
}
```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

9. Quale output si ottiene invocando il metodo p?

```
class G {  
    private Boolean b1 = new Boolean(false);  
    private String s2 = new String("");  
    private String s3;  
    void p() {  
        m(b1, new String(""));  
    }  
    void m(Boolean b2, String s1) {  
        String s4;  
        s3 = "";  
        s4 = "";  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s3) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

10. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1

B. 2

C. 3

D. 4

E. Nessuna delle precedenti

11. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("") ;
- B. Nodo<String> n = new Nodo<String>("") ;
- C. new Nodo<Boolean>("") ;
- D. new Nodo<Boolean>(true).getInfo().length();
- E. Nessuno dei precedenti

12. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

13. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

e una variabile n di tipo Nodo tale che n.info=" ", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

16. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 17**

1. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `m = v;`
  - B. `e = (Boolean) m;`
  - C. `e = v;`
  - D. `m = (Exception) e;`
  - E. `v = e;`
- 

2. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;
```

```
C(Integer y) { this.y = y; }

public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

4. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`
  - B. `Nodo<String> n = new Nodo<String>("");`
  - C. `new Nodo<Boolean>("");`
  - D. `new Nodo<Boolean>(true).getInfo().length();`
  - E. Nessuno dei precedenti
- 

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
    }
}
```

```

        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42  
 B. 41  
 C. 4  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 6. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();  
 B. int l = n.getInfo().length();  
 C. n=new Nodo(new Integer(0));  
 D. int v = n.getInfo().intValue();  
 E. Nessuno dei precedenti
- 

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
    }
}

```

```

        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452  
 B. 352Exception in thread "main" MyExc2  
 C. 35  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 8. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;  
 B. f = (Boolean []) u;  
 C. u = (Exception []) e;  
 D. u = (Exception []) f;  
 E. Nessuno dei precedenti
- 

#### 9. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {

```

```

        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1  
 B. 2.0  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuno dei precedenti
- 

10. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x  
 B. A.this.x  
 C. A.x  
 D. super.x  
 E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        }
    }
}

```

```

        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
 B. 011  
 C. 001  
 D. 101  
 E. 010
- 

12. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x  
 B. A.this.x  
 C. A.x  
 D. super.x  
 E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {

```

```

try {
    System.out.print(2);
    throw( new MyExc3() );
}
catch( MyExc2 k ) {
    throw( new MyExc2() );
}
catch( Exception e ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
    throw( new Exception() );
}
}
}

```

- A. 23431  
B. Errore a tempo di compilazione  
C. 234Exception in thread "main" java.lang.Exception  
D. 24Exception in thread "main" java.lang.Exception  
E. Nessuna delle precedenti
- 

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1

- B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

15. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12  
B. 22  
C. 3  
D. 4  
E. Nessuno dei precedenti
- 

16. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null  
B. Un oggetto può essere inizializzato dopo la sua dichiarazione  
C. Un array possiede dei membri  
D. Ai metodi static non si applica il *dynamic method dispatch*  
E. Una classe interna può essere dichiarata private
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 18**

1. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println(_QUI_);
        }
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

2. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

3. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
```

```
        System.out.print(x);
    }
}

public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

4. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

5. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

6. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
        }

```

```

        throw( new MyExc3() );
    }
    catch( MyExc2 k ) {
        throw( new MyExc2() );
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}
}

```

- A. 23431  
B. Errore a tempo di compilazione  
C. 234Exception in thread "main" java.lang.Exception  
D. 24Exception in thread "main" java.lang.Exception  
E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42  
B. 41  
C. 4  
D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 011  
C. 001  
D. 101  
E. 010
- 

12. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;  
B. e = (Boolean) m;  
C. e = v;  
D. m = (Exception) e;  
E. v = e;
- 

13. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4

E. Nessuna delle precedenti

---

15. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

16. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 19**

1. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

2. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi *static* non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata *private*
- 

3. Quale espressione va inserita nel punto indicato per stampare l'attributo *x* ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
}
```

```
public static void main(String a[]) {
    B obj = new B();
    obj.x = 1;
    obj.set_x(2);
    obj.my_print();
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

4. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
}
public static void main(String a[]) {
    new B();
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
- B. A.this.x
- C. A.x

- D. super.x
  - E. Nessuna delle precedenti
- 

7. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;

- B. `f = (Boolean []) u;`  
 C. `u = (Exception []) e;`  
 D. `u = (Exception []) f;`  
 E. Nessuno dei precedenti
- 

10. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. `Nodo<String> n = new Nodo("");`  
 B. `Nodo<String> n = new Nodo<String>("");`  
 C. `new Nodo<Boolean>("");`  
 D. `new Nodo<Boolean>(true).getInfo().length();`  
 E. Nessuno dei precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110  
 B. 011

- C. 001  
 D. 101  
 E. 010
- 

12. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
A. 12
B. 22
C. 3
D. 4
E. Nessuno dei precedenti
```

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

14. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

15. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
- 

- C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class D1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            m();  
        }  
        catch( Exception x ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 y ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 u ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 20**

1. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

2. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;
    C(Integer y) { this.y = y; }
    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
A. 1
B. 2
C. 3
D. 4
E. Nessuna delle precedenti
```

4. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
- B. Nodo<String> n = new Nodo<String>("");
- C. new Nodo<Boolean>("");
- D. new Nodo<Boolean>(true).getInfo().length();
- E. Nessuno dei precedenti

5. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

6. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

7. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

8. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}
```

```
}  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

9. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

10. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println( _QUI_ );  
        }  
    }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431  
 B. Errore a tempo di compilazione  
 C. 234Exception in thread "main" java.lang.Exception  
 D. 24Exception in thread "main" java.lang.Exception  
 E. Nessuna delle precedenti
- 

## 12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
}

```

```

    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

A. 42  

B. 41  

C. 4  

D. Errore a tempo di compilazione  

E. Nessuna delle precedenti

```

---

## 13. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

A. 1  

B. 2  

C. Errore a tempo di compilazione  

D. Errore a tempo di esecuzione  

E. Nessuno dei precedenti

```

---

## 12. Qual è l'output di questo codice?

## 14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
}

```

```

static void n() throws Exception {
    try {
        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 3452  
B. 352Exception in thread "main" MyExc2  
C. 35  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;  
B. f = (Boolean []) u;  
C. u = (Exception []) e;  
D. u = (Exception []) f;
- 

E. Nessuno dei precedenti

---

16. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 011  
C. 001  
D. 101  
E. 010
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 21**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        } catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        } catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

2. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
```

```
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

3. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110

- B. 011
  - C. 001
  - D. 101
  - E. 010
- 

4. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

5. Qual è il risultato di questo codice:

```
class A {  
    static protected int x = 1;  
    public static int get() { return x; }  
}  
  
public class B extends A {  
    public static int get() { return x+1; }  
    public static void main(String[] s) {  
        A a = new B();  
        B b = (B) a;  
        System.out.println( a.get() + b.get() );  
    }  
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class D1 {  
    public static void main(String [] argv)  
throws Exception {  
    try {  
        m();  
    }  
    catch( Exception x ) {  
        System.out.print(1);  
    }  
    catch( MyExc2 y ) {  
        System.out.print(2);  
    }  
    catch( MyExc3 u ) {  
        System.out.print(3);  
    }  
}  
static void m() throws Exception {  
    try {  
        throw( new MyExc2() );  
    }  
    finally {  
        System.out.print(4);  
    }  
}
```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

7. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
}  
public static void main(String a[]) {  
    new B();  
}
```

- A. 1
- B. 2.0
- C. Errore a tempo di compilazione

- D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

#### 8. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length()
  - E. Nessuno dei precedenti
- 

#### 9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class C1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            n();  
        }  
        catch( MyExc3 j ) {  
            System.out.print(1);  
        }  
        finally {  
            System.out.print(2);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            System.out.print(3);  
            if (true) throw( new MyExc2() );  
            else throw( new MyExc3() );  
        }  
        catch( MyExc1 g ) {  
            System.out.print(4);  
        }  
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 10. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

#### 11. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

#### 12. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println( _QUI_ );  
        }  
    }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

#### 13. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

#### 14. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

#### 15. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

#### 16. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 22**

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 011
- C. 001
- D. 101
- E. 010

2. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;
    C(Integer y) { this.y = y; }
```

```
public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione

- C. 234Exception in thread "main" java.lang.Exception  
D. 24Exception in thread "main" java.lang.Exception  
E. Nessuna delle precedenti
- 

4. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();  
B. int l = n.getInfo().length();  
C. n=new Nodo(new Integer(0));  
D. int v = n.getInfo().intValue();  
E. Nessuno dei precedenti
- 

5. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

6. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");  
B. Nodo<String> n = new Nodo<String>("");  
C. new Nodo<Boolean>("");  
D. new Nodo<Boolean>(true).getInfo().length();  
E. Nessuno dei precedenti
- 

7. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

- A. 1  
B. 2.0  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuno dei precedenti
- 

8. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

- A. x  
B. A.this.x  
C. A.x

D. super.x

E. Nessuna delle precedenti

---

9. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

---

10. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

---

11. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

---

12. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452  
B. 352Exception in thread "main" MyExc2  
C. 35  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

15. Qual è il risultato di questo codice:

---

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12  
B. 22  
C. 3  
D. 4  
E. Nessuno dei precedenti
- 

16. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x  
B. A.this.x  
C. A.x  
D. super.x  
E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 23**

1. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

2. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
- 

D. super.x

E. Nessuna delle precedenti

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

5. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

6. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {

```

```

        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

7. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

9. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println( _QUI_ );  
        }  
    }  
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class D1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            m();  
        }  
        catch( Exception x ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 y ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 u ) {  
            System.out.print(3);  
        }  
    }  
  
    static void m() throws Exception {  
        try {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

12. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

13. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
```

```
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

A. 23431
B. Errore a tempo di compilazione
C. 234Exception in thread "main" java.lang.Exception
D. 24Exception in thread "main" java.lang.Exception
E. Nessuna delle precedenti
```

---

16. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 24**

1. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

2. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

4. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
}
public static void main(String a[]) {
    new B();
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

5. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42
- B. 41
- C. 4

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

#### 8. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

#### 9. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
- B. Nodo<String> n = new Nodo<String>("");
- C. new Nodo<Boolean>("");
- D. new Nodo<Boolean>(true).getInfo().length();
- E. Nessuno dei precedenti

---

#### 10. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

---

#### 11. Quale output si ottiene invocando il metodo p?

```
class G {  
    private Boolean b1 = new Boolean(false);  
    private String s2 = new String("");  
    private String s3;  
    void p() {  
        m(b1, new String(""));  
    }  
    void m(Boolean b2, String s1) {  
        String s4;  
        s3 = "";  
        s4 = "";  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s3) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

A. 110

B. 011

C. 001

D. 101

E. 010

---

#### 12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
            System.out.print(1);  
        }  
        catch( MyExc3 h ) {  
            throw( new MyExc2() );  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 k ) {  
            throw( new MyExc2() );  
        }  
        catch( Exception e ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
    }  
}
```

A. 23431

B. Errore a tempo di compilazione

C. 234Exception in thread "main" java.lang.Exception

D. 24Exception in thread "main" java.lang.Exception

E. Nessuna delle precedenti

---

#### 13. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
}  
public class B {  
    protected int x = 2;  
    public void my_print() {
```

```

        System.out.println( _QUI_ );
    }
}

```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
}

```

```

static void n() throws Exception {
    try {
        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 25**

1. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

2. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
- 

- E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

4. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
 B. 011  
 C. 001  
 D. 101  
 E. 010
- 

5. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;  
 B. e = (Boolean) m;  
 C. e = v;  
 D. m = (Exception) e;  
 E. v = e;
- 

6. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

7. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
}
public static void main(String a[]) {
    new B();
}

```

- A. 1  
 B. 2.0  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuno dei precedenti
- 

8. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

```

    }
}

A. x
B. A.this.x
C. A.x
D. super.x
E. Nessuna delle precedenti

```

---

9. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

10. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

11. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

è una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

12. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

13. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi **static** non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata **private**
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

### 15. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("") ;
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

### 16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 26**

1. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
```

```
        n();
    }
    catch( MyExc3 j ) {
        System.out.print(1);
    }
    finally {
        System.out.print(2);
    }
}
static void n() throws Exception {
    try {
        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}
```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
```

```

        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1  
 B. 2.0  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuno dei precedenti
- 

#### 4. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");  
 B. Nodo<String> n = new Nodo<String>("");  
 C. new Nodo<Boolean>("");  
 D. new Nodo<Boolean>(true).getInfo().length();  
 E. Nessuno dei precedenti
- 

#### 5. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

#### 6. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12  
 B. 22  
 C. 3  
 D. 4  
 E. Nessuno dei precedenti
- 

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42  
 B. 41  
 C. 4  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) { try {
        p();
        System.out.print(1);
    } catch( MyExc3 h ) {
        throw( new MyExc2() );
    }
}
static void p() {
    try {
        System.out.print(2);
        throw( new MyExc3() );
    } catch( MyExc2 k ) {
        throw( new MyExc2() );
    } catch( Exception e ) {
        System.out.print(3);
    } finally {
        System.out.print(4);
        throw( new Exception() );
    }
}
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

9. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2

C. 3

D. 4

E. Nessuna delle precedenti

---

10. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

11. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

12. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;

- C. `u = (Exception []) e;`  
D. `u = (Exception []) f;`  
E. Nessuno dei precedenti

---

13. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

e una variabile n di tipo Nodo tale che `n.info=""`, dire quale dei loro usi riportati qui sotto è corretto:

- A. `String s = n.getInfo();`  
B. `int l = n.getInfo().length();`  
C. `n=new Nodo(new Integer(0));`  
D. `int v = n.getInfo().intValue();`  
E. Nessuno dei precedenti

---

14. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

- A. `x`  
B. `A.this.x`  
C. `A.x`  
D. `super.x`  
E. Nessuna delle precedenti

---

15. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println( _QUI_ );  
        }  
    }  
}
```

- A. `x`  
B. `A.this.x`  
C. `A.x`  
D. `super.x`  
E. Nessuna delle precedenti

---

16. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null  
B. Un oggetto può essere inizializzato dopo la sua dichiarazione  
C. Un array possiede dei membri  
D. Ai metodi static non si applica il *dynamic method dispatch*  
E. Una classe interna può essere dichiarata private



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 27**

1. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

3. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

4. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
```

```

        System.out.print(3);
    }
}

static void m() throws Exception {
    try {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 42  
 B. 41  
 C. 4  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

6. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

7. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452  
 B. 352Exception in thread "main" MyExc2  
 C. 35  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

9. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
  - B. 0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

10. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
    }
}

```

```

        System.out.print(1);
    }
    catch( MyExc3 h ) {
        throw( new MyExc2() );
    }
}
static void p() {
    try {
        System.out.print(2);
        throw( new MyExc3() );
    }
    catch( MyExc2 k ) {
        throw( new MyExc2() );
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}

```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

12. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

13. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

14. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println( _QUI_ );  
        }  
    }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

15. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
- 

- B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

16. Quale output si ottiene invocando il metodo p?

```
class G {  
    private Boolean b1 = new Boolean(false);  
    private String s2 = new String("");  
    private String s3;  
    void p() {  
        m(b1, new String "");  
    }  
    void m(Boolean b2, String s1) {  
        String s4;  
        s3 = "";  
        s4 = "";  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s3) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 28**

1. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

2. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
A. x
B. A.this.x
C. A.x
D. super.x
E. Nessuna delle precedenti
```

4. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

5. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info=" ", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;
}

```

```

C(Integer y) { this.y = y; }

public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

10. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

11. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

---

12. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
}
public static void main(String a[]) {
    new B();
}

```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

13. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

14. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi **static** non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata **private**
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
```

```
        throw( new MyExc2() );
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

---

16. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
A. 12
B. 22
C. 3
D. 4
E. Nessuno dei precedenti
```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 29**

1. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

2. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
    }
}
```

```
catch( Exception x ) {
    System.out.print(1);
}
catch( MyExc2 y ) {
    System.out.print(2);
}
catch( MyExc3 u ) {
    System.out.print(3);
}
}
static void m() throws Exception {
    try {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(4);
    }
}
}
```

- A. 42
- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {

```

```

        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 3452  
B. 352Exception in thread "main" MyExc2  
C. 35  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

5. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

6. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");

- B. Nodo<String> n = new Nodo<String>("")  
C. new Nodo<Boolean>("");  
D. new Nodo<Boolean>(true).getInfo().length()  
E. Nessuno dei precedenti
- 

7. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;  
B. f = (Boolean []) u;  
C. u = (Exception []) e;  
D. u = (Exception []) f;  
E. Nessuno dei precedenti
- 

8. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x  
B. A.this.x  
C. A.x  
D. super.x  
E. Nessuna delle precedenti
- 

9. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

10. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12  
 B. 22  
 C. 3  
 D. 4  
 E. Nessuno dei precedenti
- 

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x

- B. A.this.x  
 C. A.x  
 D. super.x  
 E. Nessuna delle precedenti
- 

12. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
}
public static void main(String a[]) {
    new B();
}

```

- A. 1  
 B. 2.0  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuno dei precedenti
- 

13. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null  
 B. Un oggetto può essere inizializzato dopo la sua dichiarazione  
 C. Un array possiede dei membri  
 D. Ai metodi `static` non si applica il *dynamic method dispatch*  
 E. Una classe interna può essere dichiarata `private`
- 

14. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {

```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110  
 B. 011  
 C. 001  
 D. 101  
 E. 010
- 

#### 15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {

```

```

            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431  
 B. Errore a tempo di compilazione  
 C. 234Exception in thread "main" java.lang.Exception  
 D. 24Exception in thread "main" java.lang.Exception  
 E. Nessuna delle precedenti
- 

#### 16. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1  
 B. 2  
 C. Errore a tempo di compilazione  
 D. Errore a tempo di esecuzione  
 E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 30**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;
    C(Integer y) { this.y = y; }
```

```
public static void main(String a[])
{
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;
    C(Integer y) { this.y = y; }

    public static void main(String a[])
    {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

4. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi **static** non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata **private**

---

## 5. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

## 6. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

## 7. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

A. 1

B. 2.0

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuno dei precedenti

---

## 8. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

## 9. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

A. 1

B. 2

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuno dei precedenti

---

## 10. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

**11. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
- 

**E. Nessuna delle precedenti**

**12. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

**13. Quale output si ottiene invocando il metodo p?**

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if(s2 == s1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s3) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110  
 B. 011  
 C. 001  
 D. 101  
 E. 010

14. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;  
 B. e = (Boolean) m;  
 C. e = v;  
 D. m = (Exception) e;  
 E. v = e;

15. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
```

```

    }
    public class B extends A {
        protected int x;
        public void my_print() {
            System.out.println( _QUI_ );
        }
        public static void main(String a[]) {
            B obj = new B();
            obj.x = 1;
            obj.set_x(2);
            obj.my_print();
        }
    }
}
```

- A. x  
 B. A.this.x  
 C. A.x  
 D. super.x  
 E. Nessuna delle precedenti

16. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12  
 B. 22  
 C. 3  
 D. 4  
 E. Nessuno dei precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 31**

1. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

- B. 011  
C. 001  
D. 101  
E. 010
- 

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;  
B. e = (Boolean) m;  
C. e = v;  
D. m = (Exception) e;  
E. v = e;
- 

2. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 110

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;  
B. f = (Boolean []) u;  
C. u = (Exception []) e;  
D. u = (Exception []) f;  
E. Nessuno dei precedenti

- 
5. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class D1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            m();  
        }  
        catch( Exception x ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 y ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 u ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. 42  
B. 41  
C. 4

- D. Errore a tempo di compilazione

- E. Nessuna delle precedenti
- 

7. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");  
B. Nodo<String> n = new Nodo<String>("");  
C. new Nodo<Boolean>("");  
D. new Nodo<Boolean>(true).getInfo().length()  
E. Nessuno dei precedenti
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
            System.out.print(1);  
        }  
        catch( MyExc3 h ) {  
            throw( new MyExc2() );  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 k ) {  
            throw( new MyExc2() );  
        }  
        catch( Exception e ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
    }  
}
```

- A. 23431  
B. Errore a tempo di compilazione  
C. 234Exception in thread "main" java.lang.Exception

- D. 24Exception in thread "main" java.lang.Exception  
E. Nessuna delle precedenti
- 

9. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

- A. 1  
B. 2.0  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuno dei precedenti
- 

10. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1  
B. 2  
C. Errore a tempo di compilazione  
D. Errore a tempo di esecuzione  
E. Nessuno dei precedenti
- 

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

A. x  
B. A.this.x  
C. A.x  
D. super.x  
E. Nessuna delle precedenti

---

12. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println( _QUI_ );  
        }  
    }  
}
```

A. x  
B. A.this.x  
C. A.x  
D. super.x  
E. Nessuna delle precedenti

---

13. Qual è il risultato di questo codice:

```
class A {  
    static protected int x = 1;  
    public static int get() { return x; }  
}  
public class B extends A {  
    public static int get() { return x+1; }  
    public static void main(String[] s) {  
        A a = new B();  
        B b = (B) a;  
        System.out.println( a.get() + b.get() );  
    }  
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

15. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

---

16. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 32**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

4. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi **static** non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata **private**

---

5. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

6. Qual è il risultato di questo codice:

```
class A {  
    static protected int x = 1;  
    public static int get() { return x; }  
}  
public class B extends A {  
    public static int get() { return x+1; }  
    public static void main(String[] s) {  
        A a = new B();  
        B b = (B) a;  
        System.out.println( a.get() + b.get() );  
    }  
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
            System.out.print(1);  
        }
```

```
            catch( MyExc3 h ) {  
                throw( new MyExc2() );  
            }  
        }  
        static void p() {  
            try {  
                System.out.print(2);  
                throw( new MyExc3() );  
            }  
            catch( MyExc2 k ) {  
                throw( new MyExc2() );  
            }  
            catch( Exception e ) {  
                System.out.print(3);  
            }  
            finally {  
                System.out.print(4);  
                throw( new Exception() );  
            }  
        }  
    }  
    A. 23431  
    B. Errore a tempo di compilazione  
    C. 234Exception in thread "main" java.lang.Exception  
    D. 24Exception in thread "main" java.lang.Exception  
    E. Nessuna delle precedenti
```

---

8. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class C1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            n();  
        }  
        catch( MyExc3 j ) {  
            System.out.print(1);  
        }  
    }  
}
```

```

}
finally {
    System.out.print(2);
}
}

static void n() throws Exception {
try {
    System.out.print(3);
    if (true) throw( new MyExc2() );
    else throw( new MyExc3() );
}
catch( MyExc1 g ) {
    System.out.print(4);
}
catch( MyExc3 e ) {
    throw( new MyExc1() );
}
finally {
    System.out.print(5);
}
}
}

```

- A. 3452  
B. 352Exception in thread "main" MyExc2  
C. 35  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

10. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

11. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

A. 1
B. 2
C. 3
D. 4
E. Nessuna delle precedenti

```

---

12. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;  
B. f = (Boolean []) u;  
C. u = (Exception []) e;  
D. u = (Exception []) f;  
E. Nessuno dei precedenti
- 

13. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

A. x
B. A.this.x
C. A.x
D. super.x
E. Nessuna delle precedenti

```

---

14. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
- B. Nodo<String> n = new Nodo<String>("");
- C. new Nodo<Boolean>("");
- D. new Nodo<Boolean>(true).getInfo().length();
- E. Nessuno dei precedenti

15. Quale output si ottiene invocando il metodo p?

```
class G {  
    private Boolean b1 = new Boolean(false);  
    private String s2 = new String("");  
    private String s3;  
    void p() {  
        m(b1, new String(""));  
    }  
    void m(Boolean b2, String s1) {  
        String s4;  
        s3 = "";  
        s4 = "";  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s3) {
```

```
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }
```

A. 110

B. 011

C. 001

D. 101

E. 010

---

16. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

A. 1

B. 2.0

C. Errore a tempo di compilazione

D. Errore a tempo di esecuzione

E. Nessuno dei precedenti

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 33**

1. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

2. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;
    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
- B. 2
- C. 3

D. 4

E. Nessuna delle precedenti

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        } catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        } catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

## 5. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

## 6. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println(_QUI_);
        }
    }
}

```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

## 7. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

## 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

## 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

11. Quale espressione va inserita nel punto indicato per stampare l'attributo `x` ereditato dalla classe A?

```

class A {
    protected int x;
}

```

```

    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}
A. x
B. A.this.x
C. A.x
D. super.x
E. Nessuna delle precedenti

```

---

12. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
A. 1
B. 2
C. Errore a tempo di compilazione
D. Errore a tempo di esecuzione
E. Nessuno dei precedenti

```

---

13. Date le dichiarazioni:

```

Boolean e;
Exception m;
Object v;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

---

14. Qual è il risultato di questo codice:

```
class A {  
    static protected int x = 1;  
    public static int get() { return x; }  
}  
public class B extends A {  
    public static int get() { return x+1; }  
    public static void main(String[] s) {  
        A a = new B();  
        B b = (B) a;  
        System.out.println( a.get() + b.get() );  
    }  
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

15. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
- 

D. 4

E. Nessuna delle precedenti

---

16. Quale output si ottiene invocando il metodo p?

```
class G {  
    private Boolean b1 = new Boolean(false);  
    private String s2 = new String("");  
    private String s3;  
    void p() {  
        m(b1, new String "");  
    }  
    void m(Boolean b2, String s1) {  
        String s4;  
        s3 = "";  
        s4 = "";  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s3) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 34**

1. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

2. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

3. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

4. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

5. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12

- B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

6. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

---

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class C1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            n();  
        }  
        catch( MyExc3 j ) {
```

---

```
        System.out.print(1);  
    }  
    finally {  
        System.out.print(2);  
    }  
}  
static void n() throws Exception {  
    try {  
        System.out.print(3);  
        if (true) throw( new MyExc2() );  
        else throw( new MyExc3() );  
    }  
    catch( MyExc1 g ) {  
        System.out.print(4);  
    }  
    catch( MyExc3 e ) {  
        throw( new MyExc1() );  
    }  
    finally {  
        System.out.print(5);  
    }  
}
```

---

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

---

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

#### 11. Date la classe

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

#### 12. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

A. 1
B. 2
C. 3
D. 4
E. Nessuna delle precedenti

```

---

#### 13. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

A. 1
B. 2
C. 3
D. 4
E. Nessuna delle precedenti

```

---

#### 14. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
    }
}

```

```

if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s2 == s1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 110  
B. 011  
C. 001  
D. 101  
E. 010
- 

```

        catch( MyExc3 u ) {
            System.out.print(3);
        }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42  
B. 41  
C. 4  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

**15. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
    }
}

```

**16. Date le dichiarazioni:**

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;  
B. f = (Boolean []) u;  
C. u = (Exception []) e;  
D. u = (Exception []) f;  
E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 35**

1. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

2. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
```

```
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

3. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

4. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

5. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

6. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println(_QUI_ );  
        }  
    }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1

- B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

9. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

10. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println(_QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

12. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;
    C(Integer y) { this.y = y; }
```

```
public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
A. 1
B. 2
C. 3
D. 4
E. Nessuna delle precedenti
```

---

13. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
}
public static void main(String a[]) {
    new B();
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
}
```

```

    }
}

static void m() throws Exception {
    try {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. 42  
 B. 41  
 C. 4  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
        }
    }
}

```

```

        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 3452  
 B. 352Exception in thread "main" MyExc2  
 C. 35  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;  
 B. f = (Boolean []) u;  
 C. u = (Exception []) e;  
 D. u = (Exception []) f;  
 E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 36**

1. Quale risultato produce questo codice?

```
class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

2. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi `static` non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata `private`
- 

3. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

4. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

5. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
    }
}
```

```

        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

6. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

7. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

8. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
    }
}

```

```

        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431  
B. Errore a tempo di compilazione  
C. 234Exception in thread "main" java.lang.Exception  
D. 24Exception in thread "main" java.lang.Exception  
E. Nessuna delle precedenti
- 

10. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

11. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

```

- A. x  
B. A.this.x

- C. A.x  
D. super.x  
E. Nessuna delle precedenti
- 

12. Qual è il risultato di questo codice:

```

class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}

```

- A. 12  
B. 22  
C. 3  
D. 4  
E. Nessuno dei precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {

```

```

        System.out.print(5);
    }
}

```

- A. 3452  
 B. 352Exception in thread "main" MyExc2  
 C. 35  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

**14. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. 42  
 B. 41

C. 4

- D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

**15. Date la classe**

```

class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}

```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();  
 B. int l = n.getInfo().length();  
 C. n=new Nodo(new Integer(0));  
 D. int v = n.getInfo().intValue();  
 E. Nessuno dei precedenti
- 

**16. Date le dichiarazioni:**

```

Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;  
 B. f = (Boolean []) u;  
 C. u = (Exception []) e;  
 D. u = (Exception []) f;  
 E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 37**

1. Quale risultato produce questo codice?

```
class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

2. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    static Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

3. Date la classe

```
class Nodo {
    private Object info;
    Nodo next;
    Nodo(Object i) { info=i; }
    public Object getInfo() { return info; }
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
  - B. int l = n.getInfo().length();
  - C. n=new Nodo(new Integer(0));
  - D. int v = n.getInfo().intValue();
  - E. Nessuno dei precedenti
- 

4. Date le dichiarazioni:

```
Boolean e;
Exception m;
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

5. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

6. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

7. Qual è il risultato di questo codice:

```
class A {
    static protected int x = 1;
    public static int get() { return x; }
}
public class B extends A {
    public static int get() { return x+1; }
    public static void main(String[] s) {
        A a = new B();
        B b = (B) a;
        System.out.println( a.get() + b.get() );
    }
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;
    C(Integer y) { this.y = y; }
```

```
public static void main(String a[]) {
    C obj = new C( new Integer(1) );
    obj = null; // QUESTA
    System.out.println("done");
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

11. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
- 

12. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
}
public static void main(String a[]) {
    new B();
}

```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. 3452
  - B. 352Exception in thread "main" MyExc2
  - C. 35
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Quale output si ottiene invocando il metodo p?

```

class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

if(s2 == s1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s3) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}

```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

15. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public void my_print() {
        System.out.println( _QUI_ );
    }
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        obj.my_print();
    }
}

```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
- 

E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 38**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

3. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[])
        {
            C obj = new C( new Integer(1) );
            obj = null; // QUESTA
            System.out.println("done");
        }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

4. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}
```

- A. x

- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

---

5. Qual è il risultato di questo codice:

```
class A {  
    static protected int x = 1;  
    public static int get() { return x; }  
}  
public class B extends A {  
    public static int get() { return x+1; }  
    public static void main(String[] s) {  
        A a = new B();  
        B b = (B) a;  
        System.out.println( a.get() + b.get() );  
    }  
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

---

6. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

---

7. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*

- E. Una classe interna può essere dichiarata private

---

8. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}  
  
A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
```

---

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class C1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            n();  
        }  
        catch( MyExc3 j ) {  
            System.out.print(1);  
        }  
        finally {  
            System.out.print(2);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            System.out.print(3);  
            if (true) throw( new MyExc2() );  
            else throw( new MyExc3() );  
        }  
        catch( MyExc1 g ) {  
            System.out.print(4);  
        }  
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

```

    }
}

A. 3452
B. 352Exception in thread "main" MyExc2
C. 35
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

10. Quale risultato produce questo codice?

```

class A {
    int x;
    A() {
        x = 1;
        System.out.print(x);
    }
}
public class B extends A {
    float x;
    B() {
        x = 2;
        System.out.print(x);
    }
    public static void main(String a[]) {
        new B();
    }
}

```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

```

    }
    catch( MyExc2 k ) {
        throw( new MyExc2() );
    }
    catch( Exception e ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new Exception() );
    }
}
}
A. 23431
B. Errore a tempo di compilazione
C. 234Exception in thread "main" java.lang.Exception
D. 24Exception in thread "main" java.lang.Exception
E. Nessuna delle precedenti

```

---

12. Dato il template

```

class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}

```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length();
  - E. Nessuno dei precedenti
- 

13. Quale risultato produce questo codice?

```

class A {
    protected int x;
    public void set_x(int v) { x=v; }
}
public class B extends A {
    protected int x;
    public static void main(String a[]) {
        B obj = new B();
        obj.x = 1;
        obj.set_x(2);
        System.out.println( obj.x );
    }
}

```

- A. 1
- B. 2
- C. Errore a tempo di compilazione

- D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

14. Quale output si ottiene invocando il metodo p?

```
class G {  
    private Boolean b1 = new Boolean(false);  
    private String s2 = new String("");  
    private String s3;  
    void p() {  
        m(b1, new String(""));  
    }  
    void m(Boolean b2, String s1) {  
        String s4;  
        s3 = "";  
        s4 = "";  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s3) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

15. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

---

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 39**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            m();
        }
        catch( Exception x ) {
            System.out.print(1);
        }
        catch( MyExc2 y ) {
            System.out.print(2);
        }
        catch( MyExc3 u ) {
            System.out.print(3);
        }
    }
    static void m() throws Exception {
        try {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. 42
  - B. 41
  - C. 4
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Object [] e;
Boolean [] f;
Exception [] u;
u = new Exception [8];
e = new Exception [1];
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
  - B. f = (Boolean []) u;
  - C. u = (Exception []) e;
  - D. u = (Exception []) f;
  - E. Nessuno dei precedenti
- 

3. Quale output si ottiene invocando il metodo p?

```
class G {
    private Boolean b1 = new Boolean(false);
    private String s2 = new String("");
    private String s3;
    void p() {
        m(b1, new String(""));
    }
    void m(Boolean b2, String s1) {
        String s4;
        s3 = "";
        s4 = "";
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s2 == s1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s3) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
-

4. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

- A. x
- B. A.this.x
- C. A.x
- D. super.x
- E. Nessuna delle precedenti

5. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        System.out.println( obj.x );  
    }  
}
```

- A. 1
- B. 2
- C. Errore a tempo di compilazione
- D. Errore a tempo di esecuzione
- E. Nessuno dei precedenti

6. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

7. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
- B. e = (Boolean) m;
- C. e = v;
- D. m = (Exception) e;
- E. v = e;

8. Qual è il risultato di questo codice:

```
class A {  
    static protected int x = 1;  
    public static int get() { return x; }  
}  
public class B extends A {  
    public static int get() { return x+1; }  
    public static void main(String[] s) {  
        A a = new B();  
        B b = (B) a;  
        System.out.println( a.get() + b.get() );  
    }  
}
```

- A. 12
- B. 22
- C. 3
- D. 4
- E. Nessuno dei precedenti

9. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```

class A {
    int x = 1;

    public class B {
        protected int x = 2;
        public void my_print() {
            System.out.println( _QUI_ );
        }
    }
}

A. x
B. A.this.x
C. A.x
D. super.x
E. Nessuna delle precedenti

```

---

10. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```

public class C {
    String x = "abc";
    Integer y;

    C(Integer y) { this.y = y; }

    public static void main(String a[]) {
        C obj = new C( new Integer(1) );
        obj = null; // QUESTA
        System.out.println("done");
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti

- 
11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
    throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
}

```

```

    }
}
static void n() throws Exception {
    try {
        System.out.print(3);
        if (true) throw( new MyExc2() );
        else throw( new MyExc3() );
    }
    catch( MyExc1 g ) {
        System.out.print(4);
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. 3452  
B. 352Exception in thread "main" MyExc2  
C. 35  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

- 
12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}

```

- A. 23431
  - B. Errore a tempo di compilazione
  - C. 234Exception in thread "main" java.lang.Exception
  - D. 24Exception in thread "main" java.lang.Exception
  - E. Nessuna delle precedenti
- 

13. Dato il template

```
class Nodo<T> {  
    private T info;  
    Nodo<T> next;  
    Nodo(T i) { info=i; }  
    public T getInfo() { return info; }  
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");
  - B. Nodo<String> n = new Nodo<String>("");
  - C. new Nodo<Boolean>("");
  - D. new Nodo<Boolean>(true).getInfo().length(); }
  - E. Nessuno dei precedenti
- 

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
- B. 2

- C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

15. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

16. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
  - B. Un oggetto può essere inizializzato dopo la sua dichiarazione
  - C. Un array possiede dei membri
  - D. Ai metodi static non si applica il *dynamic method dispatch*
  - E. Una classe interna può essere dichiarata private
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 40**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
        catch( MyExc3 j ) {
            System.out.print(1);
        }
        finally {
            System.out.print(2);
        }
    }
    static void n() throws Exception {
        try {
            System.out.print(3);
            if (true) throw( new MyExc2() );
            else throw( new MyExc3() );
        }
        catch( MyExc1 g ) {
            System.out.print(4);
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. 3452
- B. 352Exception in thread "main" MyExc2
- C. 35
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
```

```
class MyExc3 extends Exception { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
            System.out.print(1);
        }
        catch( MyExc3 h ) {
            throw( new MyExc2() );
        }
    }
    static void p() {
        try {
            System.out.print(2);
            throw( new MyExc3() );
        }
        catch( MyExc2 k ) {
            throw( new MyExc2() );
        }
        catch( Exception e ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new Exception() );
        }
    }
}
```

- A. 23431
- B. Errore a tempo di compilazione
- C. 234Exception in thread "main" java.lang.Exception
- D. 24Exception in thread "main" java.lang.Exception
- E. Nessuna delle precedenti

3. Dato il template

```
class Nodo<T> {
    private T info;
    Nodo<T> next;
    Nodo(T i) { info=i; }
    public T getInfo() { return info; }
}
```

dire quale dei suoi usi riportati qui sotto è corretto:

- A. Nodo<String> n = new Nodo("");

- B. Nodo<String> n = new Nodo<String>("");
- C. new Nodo<Boolean>("");
- D. new Nodo<Boolean>(true).getInfo().length();
- E. Nessuno dei precedenti

4. Date la classe

```
class Nodo {  
    private Object info;  
    Nodo next;  
    Nodo(Object i) { info=i; }  
    public Object getInfo() { return info; }  
}
```

e una variabile n di tipo Nodo tale che n.info="", dire quale dei loro usi riportati qui sotto è corretto:

- A. String s = n.getInfo();
- B. int l = n.getInfo().length();
- C. n=new Nodo(new Integer(0));
- D. int v = n.getInfo().intValue();
- E. Nessuno dei precedenti

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class D1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            m();  
        }  
        catch( Exception x ) {  
            System.out.print(1);  
        }  
        catch( MyExc2 y ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 u ) {  
            System.out.print(3);  
        }  
    }  
    static void m() throws Exception {  
        try {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

A. 42

- B. 41
- C. 4
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

6. Dire quale delle seguenti affermazioni è falsa:

- A. Un array vuoto può avere riferimento null
- B. Un oggetto può essere inizializzato dopo la sua dichiarazione
- C. Un array possiede dei membri
- D. Ai metodi static non si applica il *dynamic method dispatch*
- E. Una classe interna può essere dichiarata private

7. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    static Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
Object [] e;  
Boolean [] f;  
Exception [] u;  
u = new Exception [8];  
e = new Exception [1];  
f = new Boolean [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. f = (Boolean []) e;
- B. f = (Boolean []) u;
- C. u = (Exception []) e;
- D. u = (Exception []) f;
- E. Nessuno dei precedenti

---

9. Qual è il risultato di questo codice:

```
class A {  
    static protected int x = 1;  
    public static int get() { return x; }  
}  
public class B extends A {  
    public static int get() { return x+1; }  
    public static void main(String[] s) {  
        A a = new B();  
        B b = (B) a;  
        System.out.println( a.get() + b.get() );  
    }  
}
```

- A. 12
  - B. 22
  - C. 3
  - D. 4
  - E. Nessuno dei precedenti
- 

10. Quale espressione va inserita nel punto indicato per stampare l'attributo x ereditato dalla classe A?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public void my_print() {  
        System.out.println( _QUI_ );  
    }  
    public static void main(String a[]) {  
        B obj = new B();  
        obj.x = 1;  
        obj.set_x(2);  
        obj.my_print();  
    }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```
class G {  
    private Boolean b1 = new Boolean(false);  
    private String s2 = new String("");  
    private String s3;  
    void p() {
```

```
        m(b1, new String(""));  
    }  
    void m(Boolean b2, String s1) {  
        String s4;  
        s3 = "";  
        s4 = "";  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s2 == s1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s3) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
  - B. 011
  - C. 001
  - D. 101
  - E. 010
- 

12. Date le dichiarazioni:

```
Boolean e;  
Exception m;  
Object v;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. m = v;
  - B. e = (Boolean) m;
  - C. e = v;
  - D. m = (Exception) e;
  - E. v = e;
- 

13. Quale espressione va inserita nel punto indicato per stampare l'attributo x dichiarato nella classe A?

```
class A {  
    int x = 1;  
  
    public class B {  
        protected int x = 2;  
        public void my_print() {  
            System.out.println( _QUI_ );  
        }  
    }  
}
```

- A. x
  - B. A.this.x
  - C. A.x
  - D. super.x
  - E. Nessuna delle precedenti
- 

14. Quanti oggetti diventano eleggibili per il garbage collector a causa dell'istruzione indicata?

```
public class C {  
    String x = "abc";  
    Integer y;  
  
    C(Integer y) { this.y = y; }  
  
    public static void main(String a[]) {  
        C obj = new C( new Integer(1) );  
        obj = null; // QUESTA  
        System.out.println("done");  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

15. Quale risultato produce questo codice?

```
class A {  
    protected int x;  
    public void set_x(int v) { x=v; }  
}  
public class B extends A {  
    protected int x;  
    public static void main(String a[]) {  
        B obj = new B();  
    }  
}
```

```
    obj.x = 1;  
    obj.set_x(2);  
    System.out.println( obj.x );  
}  
}
```

- A. 1
  - B. 2
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
- 

16. Quale risultato produce questo codice?

```
class A {  
    int x;  
    A() {  
        x = 1;  
        System.out.print(x);  
    }  
}  
public class B extends A {  
    float x;  
    B() {  
        x = 2;  
        System.out.print(x);  
    }  
    public static void main(String a[]) {  
        new B();  
    }  
}
```

- A. 1
  - B. 2.0
  - C. Errore a tempo di compilazione
  - D. Errore a tempo di esecuzione
  - E. Nessuno dei precedenti
-

**Prova n. 1**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 2**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 3**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 4**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 5**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 6**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 11**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 16**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

1 marzo 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   | ■ | ■ |   |   |
| 3  |   |   |   |   |   | ■ | ■ |
| 4  |   |   | ■ |   |   |   |   |
| 5  |   |   |   | ■ | ■ |   |   |
| 6  |   |   |   |   |   |   | ■ |
| 7  | ■ | ■ |   |   |   |   |   |
| 8  |   |   |   |   |   |   | ■ |
| 9  |   |   |   | ■ | ■ |   |   |
| 10 | ■ | ■ |   |   |   |   |   |
| 11 |   |   | ■ |   |   |   |   |
| 12 |   |   |   |   | ■ | ■ |   |
| 13 |   |   |   |   |   | ■ | ■ |
| 14 | ■ | ■ |   |   |   |   |   |
| 15 |   |   |   | ■ | ■ |   |   |
| 16 |   | ■ |   |   |   |   |   |

Risultato prova n. 1:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 2:

|    |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|--|
| 1  |   | ■ |   |   |   |   |  |
| 2  |   |   | ■ |   |   |   |  |
| 3  |   |   |   | ■ |   |   |  |
| 4  |   |   |   |   | ■ |   |  |
| 5  |   |   |   |   | ■ |   |  |
| 6  |   |   |   |   | ■ |   |  |
| 7  |   |   |   |   |   | ■ |  |
| 8  |   |   | ■ |   |   |   |  |
| 9  |   |   |   |   |   | ■ |  |
| 10 |   |   |   |   | ■ |   |  |
| 11 | ■ | ■ |   |   |   |   |  |
| 12 | ■ | ■ |   |   |   |   |  |
| 13 |   |   | ■ |   |   |   |  |
| 14 |   |   |   |   |   | ■ |  |
| 15 | ■ | ■ |   |   |   |   |  |
| 16 |   |   |   | ■ |   |   |  |

Risultato prova n. 3:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 4:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   |   |   |   | ■ |
| 3  |  |   |   | ■ | ■ |   |   |
| 4  |  |   | ■ | ■ |   |   |   |
| 5  |  |   |   |   | ■ | ■ |   |
| 6  |  |   | ■ | ■ |   |   |   |
| 7  |  |   |   |   |   | ■ | ■ |
| 8  |  | ■ | ■ |   |   |   |   |
| 9  |  |   |   | ■ | ■ |   |   |
| 10 |  |   |   | ■ | ■ |   |   |
| 11 |  |   |   | ■ | ■ |   |   |
| 12 |  |   |   |   | ■ | ■ |   |
| 13 |  |   |   |   | ■ | ■ |   |
| 14 |  | ■ | ■ |   |   |   |   |
| 15 |  | ■ | ■ |   |   |   |   |
| 16 |  |   | ■ |   |   |   |   |

Risultato prova n. 5:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  |   |   |   |   |   |
| 2  |   |   |   |   |   |
| 3  |   | ■ |   |   |   |
| 4  |   |   |   |   | ■ |
| 5  |   |   |   | ■ |   |
| 6  |   |   |   |   | ■ |
| 7  |   |   |   |   | ■ |
| 8  |   | ■ |   |   |   |
| 9  |   |   |   |   | ■ |
| 10 |   | ■ |   |   |   |
| 11 |   |   |   |   | ■ |
| 12 |   |   | ■ |   |   |
| 13 |   |   | ■ |   |   |
| 14 |   |   |   | ■ |   |
| 15 |   |   | ■ |   |   |
| 16 | ■ |   |   |   |   |

Risultato prova n. 6:

|    |   |  |   |   |   |   |
|----|---|--|---|---|---|---|
| 1  |   |  |   |   |   |   |
| 2  |   |  | ■ |   |   |   |
| 3  |   |  |   | ■ | ■ |   |
| 4  |   |  |   | ■ |   |   |
| 5  |   |  | ■ |   |   |   |
| 6  | ■ |  |   |   |   |   |
| 7  |   |  |   | ■ |   |   |
| 8  |   |  |   |   | ■ |   |
| 9  | ■ |  |   |   |   |   |
| 10 |   |  |   |   | ■ |   |
| 11 |   |  |   | ■ |   |   |
| 12 |   |  | ■ |   |   |   |
| 13 |   |  |   |   |   | ■ |
| 14 | ■ |  |   |   |   |   |
| 15 |   |  |   |   |   | ■ |
| 16 | ■ |  |   |   |   |   |

Risultato prova n. 7:

A 16x16 grid with red squares at the following coordinates:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (1, 5)
- (2, 5)
- (3, 5)
- (4, 5)
- (5, 5)
- (6, 5)
- (7, 5)
- (8, 5)
- (9, 5)
- (10, 5)
- (11, 5)
- (12, 5)
- (13, 5)
- (14, 5)
- (15, 5)
- (16, 5)
- (1, 6)
- (2, 6)
- (3, 6)
- (4, 6)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (1, 7)
- (2, 7)
- (3, 7)
- (4, 7)
- (5, 7)
- (6, 7)
- (7, 7)
- (8, 7)
- (9, 7)
- (10, 7)
- (11, 7)
- (12, 7)
- (13, 7)
- (14, 7)
- (15, 7)
- (16, 7)
- (1, 8)
- (2, 8)
- (3, 8)
- (4, 8)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (1, 9)
- (2, 9)
- (3, 9)
- (4, 9)
- (5, 9)
- (6, 9)
- (7, 9)
- (8, 9)
- (9, 9)
- (10, 9)
- (11, 9)
- (12, 9)
- (13, 9)
- (14, 9)
- (15, 9)
- (16, 9)
- (1, 10)
- (2, 10)
- (3, 10)
- (4, 10)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (1, 11)
- (2, 11)
- (3, 11)
- (4, 11)
- (5, 11)
- (6, 11)
- (7, 11)
- (8, 11)
- (9, 11)
- (10, 11)
- (11, 11)
- (12, 11)
- (13, 11)
- (14, 11)
- (15, 11)
- (16, 11)
- (1, 12)
- (2, 12)
- (3, 12)
- (4, 12)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (1, 13)
- (2, 13)
- (3, 13)
- (4, 13)
- (5, 13)
- (6, 13)
- (7, 13)
- (8, 13)
- (9, 13)
- (10, 13)
- (11, 13)
- (12, 13)
- (13, 13)
- (14, 13)
- (15, 13)
- (16, 13)
- (1, 14)
- (2, 14)
- (3, 14)
- (4, 14)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)
- (1, 15)
- (2, 15)
- (3, 15)
- (4, 15)
- (5, 15)
- (6, 15)
- (7, 15)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 15)
- (16, 15)
- (1, 16)
- (2, 16)
- (3, 16)
- (4, 16)
- (5, 16)
- (6, 16)
- (7, 16)
- (8, 16)
- (9, 16)
- (10, 16)
- (11, 16)
- (12, 16)
- (13, 16)
- (14, 16)
- (15, 16)
- (16, 16)

## Risultato prova n. 8:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  |   |   |   |   | ■ | ■ |   |
| 4  |   |   |   | ■ | ■ |   |   |
| 5  |   |   | ■ | ■ |   |   |   |
| 6  |   |   |   |   | ■ | ■ |   |
| 7  |   |   |   |   |   | ■ | ■ |
| 8  |   | ■ | ■ |   |   |   |   |
| 9  |   |   |   | ■ | ■ |   |   |
| 10 |   |   |   |   |   | ■ | ■ |
| 11 |   |   | ■ | ■ |   |   |   |
| 12 |   |   |   | ■ | ■ |   |   |
| 13 |   |   | ■ | ■ |   |   |   |
| 14 |   | ■ | ■ |   |   |   |   |
| 15 |   |   |   | ■ | ■ |   |   |
| 16 | ■ | ■ |   |   |   |   |   |

Risultato prova n. 9:

## Risultato prova n. 10:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 11:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 12:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 13:

|    |  |    |    |    |    |  |  |
|----|--|----|----|----|----|--|--|
| 1  |  | ■  |    |    |    |  |  |
| 2  |  |    | ■■ |    |    |  |  |
| 3  |  |    | ■■ |    |    |  |  |
| 4  |  |    |    | ■■ |    |  |  |
| 5  |  |    |    | ■■ |    |  |  |
| 6  |  |    |    |    | ■■ |  |  |
| 7  |  | ■■ |    |    |    |  |  |
| 8  |  |    | ■■ |    |    |  |  |
| 9  |  | ■■ |    |    |    |  |  |
| 10 |  |    |    | ■■ |    |  |  |
| 11 |  |    |    | ■■ |    |  |  |
| 12 |  |    |    |    | ■■ |  |  |
| 13 |  | ■■ |    |    |    |  |  |
| 14 |  |    |    |    | ■■ |  |  |
| 15 |  |    |    |    | ■■ |  |  |
| 16 |  |    |    |    | ■■ |  |  |

Risultato prova n. 14:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  | ■ | ■ |   |   |   |   |   |
| 4  |   |   |   | ■ | ■ |   |   |
| 5  |   |   |   | ■ | ■ |   |   |
| 6  |   |   | ■ | ■ |   |   |   |
| 7  |   |   |   |   | ■ | ■ |   |
| 8  |   |   |   |   |   | ■ | ■ |
| 9  |   |   | ■ | ■ |   |   |   |
| 10 | ■ | ■ |   |   |   |   |   |
| 11 |   |   |   |   | ■ | ■ |   |
| 12 |   |   |   | ■ | ■ |   |   |
| 13 |   |   |   |   | ■ | ■ |   |
| 14 |   |   |   | ■ | ■ |   |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 | ■ | ■ |   |   |   |   |   |

Risultato prova n. 15:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   | ■ |   |   |   |
| 3  |  |   |   |   |   |   | ■ |
| 4  |  |   |   |   | ■ |   |   |
| 5  |  | ■ |   |   |   |   |   |
| 6  |  |   | ■ |   |   |   |   |
| 7  |  |   |   |   |   | ■ |   |
| 8  |  | ■ |   |   |   |   |   |
| 9  |  |   |   |   |   | ■ |   |
| 10 |  |   | ■ |   |   |   |   |
| 11 |  |   |   | ■ |   |   |   |
| 12 |  |   |   |   |   | ■ |   |
| 13 |  |   |   |   | ■ |   |   |
| 14 |  | ■ |   |   |   |   |   |
| 15 |  |   |   | ■ |   |   |   |
| 16 |  |   |   |   |   |   | ■ |

Risultato prova n. 16:

|    |  |  |  |  |  |  |    |
|----|--|--|--|--|--|--|----|
| 1  |  |  |  |  |  |  | 1  |
| 2  |  |  |  |  |  |  | 2  |
| 3  |  |  |  |  |  |  | 3  |
| 4  |  |  |  |  |  |  | 4  |
| 5  |  |  |  |  |  |  | 5  |
| 6  |  |  |  |  |  |  | 6  |
| 7  |  |  |  |  |  |  | 7  |
| 8  |  |  |  |  |  |  | 8  |
| 9  |  |  |  |  |  |  | 9  |
| 10 |  |  |  |  |  |  | 10 |
| 11 |  |  |  |  |  |  | 11 |
| 12 |  |  |  |  |  |  | 12 |
| 13 |  |  |  |  |  |  | 13 |
| 14 |  |  |  |  |  |  | 14 |
| 15 |  |  |  |  |  |  | 15 |
| 16 |  |  |  |  |  |  | 16 |

Risultato prova n. 17:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 18:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 19:

|    |  |   |   |   |   |
|----|--|---|---|---|---|
| 1  |  |   |   |   |   |
| 2  |  |   |   |   |   |
| 3  |  | ■ |   |   |   |
| 4  |  |   | ■ |   |   |
| 5  |  | ■ |   |   |   |
| 6  |  |   | ■ |   |   |
| 7  |  |   |   | ■ |   |
| 8  |  |   |   |   | ■ |
| 9  |  |   |   |   | ■ |
| 10 |  |   |   | ■ |   |
| 11 |  |   | ■ |   |   |
| 12 |  |   |   | ■ |   |
| 13 |  | ■ |   |   |   |
| 14 |  | ■ |   |   |   |
| 15 |  |   | ■ |   |   |
| 16 |  |   |   | ■ |   |

Risultato prova n. 20:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 21:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 22:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   |   | ■ | ■ |   |
| 3  |  | ■ |   |   |   |   |   |
| 4  |  |   |   | ■ | ■ |   |   |
| 5  |  |   |   |   |   | ■ | ■ |
| 6  |  |   |   |   | ■ | ■ |   |
| 7  |  |   |   |   |   | ■ | ■ |
| 8  |  |   | ■ | ■ |   |   |   |
| 9  |  |   |   | ■ | ■ |   |   |
| 10 |  |   |   |   | ■ | ■ |   |
| 11 |  | ■ |   |   |   |   |   |
| 12 |  |   |   | ■ | ■ |   |   |
| 13 |  |   | ■ | ■ |   |   |   |
| 14 |  | ■ |   |   |   |   |   |
| 15 |  |   | ■ | ■ |   |   |   |
| 16 |  |   |   | ■ | ■ |   |   |

Risultato prova n. 23:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 24:

|    |   |   |   |   |
|----|---|---|---|---|
| 1  |   |   | ■ |   |
| 2  |   |   |   | ■ |
| 3  |   | ■ |   |   |
| 4  |   |   |   | ■ |
| 5  |   |   |   | ■ |
| 6  |   | ■ |   |   |
| 7  |   |   |   | ■ |
| 8  |   |   | ■ |   |
| 9  |   |   | ■ |   |
| 10 | ■ |   |   |   |
| 11 |   |   | ■ |   |
| 12 | ■ |   |   |   |
| 13 | ■ |   |   |   |
| 14 | ■ |   |   |   |
| 15 |   | ■ |   |   |
| 16 |   |   |   | ■ |

Risultato prova n. 25:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 26:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 27:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 28:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 29:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 30:

## Risultato prova n. 31:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 32:

|    |   |   |   |   |   |   |    |
|----|---|---|---|---|---|---|----|
| 1  |   |   |   |   |   |   | 1  |
| 2  |   |   | 1 |   |   |   | 2  |
| 3  |   |   | 1 |   |   |   | 3  |
| 4  |   |   |   | 1 |   |   | 4  |
| 5  |   |   | 1 |   |   |   | 5  |
| 6  |   |   |   | 1 |   |   | 6  |
| 7  |   |   |   | 1 |   |   | 7  |
| 8  |   |   |   |   | 1 |   | 8  |
| 9  | 1 | 1 |   |   |   |   | 9  |
| 10 | 1 | 1 |   |   |   |   | 10 |
| 11 |   |   |   |   | 1 |   | 11 |
| 12 | 1 | 1 |   |   |   |   | 12 |
| 13 |   |   |   |   |   | 1 | 13 |
| 14 |   |   |   | 1 | 1 |   | 14 |
| 15 | 1 | 1 |   |   |   |   | 15 |
| 16 |   |   |   |   | 1 |   | 16 |

Risultato prova n. 33:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   |   |   |   | ■ |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  |   |   |   | ■ | ■ |   |   |
| 6  |   |   |   |   | ■ | ■ |   |
| 7  | ■ | ■ |   |   |   |   |   |
| 8  | ■ | ■ |   |   |   |   |   |
| 9  |   |   |   |   |   |   | ■ |
| 10 |   |   | ■ | ■ |   |   |   |
| 11 |   |   |   | ■ | ■ |   |   |
| 12 | ■ | ■ |   |   |   |   |   |
| 13 |   |   | ■ | ■ |   |   |   |
| 14 |   |   |   |   | ■ | ■ |   |
| 15 |   |   |   |   | ■ | ■ |   |
| 16 |   |   |   | ■ | ■ |   |   |

Risultato prova n. 34:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  |   |   |   | ■ | ■ |   |   |
| 5  |   | ■ | ■ |   |   |   |   |
| 6  |   |   |   | ■ | ■ |   |   |
| 7  |   |   | ■ | ■ |   |   |   |
| 8  |   |   |   |   |   | ■ | ■ |
| 9  |   |   | ■ | ■ |   |   |   |
| 10 |   |   |   |   | ■ | ■ |   |
| 11 |   |   | ■ | ■ |   |   |   |
| 12 | ■ | ■ |   |   |   |   |   |
| 13 |   |   |   |   |   | ■ | ■ |
| 14 |   |   |   |   | ■ | ■ |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 |   |   |   | ■ | ■ |   |   |

Risultato prova n. 35:

## Risultato prova n. 36:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 37:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 38:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 39:

|    |    |    |    |    |    |    |  |
|----|----|----|----|----|----|----|--|
| 1  |    | ■  |    |    |    |    |  |
| 2  |    |    | ■■ |    |    |    |  |
| 3  |    |    | ■■ |    |    |    |  |
| 4  |    |    |    | ■■ |    |    |  |
| 5  |    |    |    |    | ■■ |    |  |
| 6  |    | ■■ |    |    |    |    |  |
| 7  |    | ■■ |    |    |    |    |  |
| 8  |    |    | ■■ |    |    |    |  |
| 9  |    |    | ■■ |    |    |    |  |
| 10 |    |    |    | ■■ |    |    |  |
| 11 |    |    |    |    | ■■ |    |  |
| 12 |    |    |    |    |    | ■■ |  |
| 13 |    |    |    | ■■ |    |    |  |
| 14 |    |    | ■■ |    |    |    |  |
| 15 | ■■ |    |    |    |    |    |  |
| 16 |    |    |    |    |    | ■■ |  |

Risultato prova n. 40:



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 1**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f();
    }
}
```

```
    c1.f(1);
}
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}
```

```

    }

    public static void main(String[] args) {
        new X().new I().print();
    }
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a))
}

```

```

    + d.f(b)
    + d.f(c) );
}

A. DDD
B. ABC
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

7. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
D. Un oggetto può non esistere dopo la sua dichiarazione  
E. Una classe non interna può essere dichiarata **private**
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}
public static void main(String args[]) {
    new Z(1);
}

A. Z
B. XYZ
C. XXYZ

```

- D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

9. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 111  
B. 010  
C. 110  
D. 100  
E. 101
- 

10. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;

- B. d = b;  
C. c = d;  
D. b = c;  
E. b = (C1) d;
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                n();  
                System.out.print(1);  
            }  
            catch( Exception w ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 z ) {  
                throw( new Exception() );  
            }  
            catch( MyExc1 u ) {  
            }  
            finally {  
                System.out.print(3);  
            }  
        }  
        static void n() throws Exception {  
            try {  
                throw( new Exception() );  
            }  
            catch( MyExc3 e ) {  
                throw( new MyExc1() );  
            }  
            finally {  
                System.out.print(4);  
            }  
        }  
    }
```

- A. Errore a tempo di compilazione  
B. 4213  
C. 41  
D. 423  
E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {
```

```

        System.out.print(1);
        q();
        System.out.print(2);
    }
    catch( MyExc2 d ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc3() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}
}

```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

### 13. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;  
D. f = (Exception []) n;  
E. Nessuno dei precedenti
- 

### 14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

### 15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}

```

```
    }
}

A. DDD
B. ABC
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.
```

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
```

```
class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}
public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 2**

1. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
```

```
try {
    n();
    System.out.print(1);
}
catch( Exception w ) {
    System.out.print(2);
}
catch( MyExc3 z ) {
    throw( new Exception() );
}
catch( MyExc1 u ) {
}
finally {
    System.out.print(3);
}
}
static void n() throws Exception {
try {
    throw( new Exception() );
}
catch( MyExc3 e ) {
    throw( new MyExc1() );
}
finally {
    System.out.print(4);
}
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
    try {
        System.out.print(1);
        q();
    }
```

```

        System.out.print(2);
    }
    catch( MyExc2 d ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc3() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
 B. XYZ

C. XXYZ

- D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
 B. XYZ  
 C. XXYZ  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

10. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
 B. f = (Exception []) z;  
 C. z = (Object [] []) f;  
 D. f = (Exception []) n;  
 E. Nessuno dei precedenti

- 
11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.

- E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
}
```

```
public static void main(String[] args) {  
    new X().new I().print();  
}
```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
        catch( MyExc3 z ) {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata private
- 

16. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 3**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
    }
}
```

```
finally {
    throw( new MyExc3() );
}
}
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}
}
```

- A. 146732Exception in thread "main" MyExc3
- B. 14666666... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 1463
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
    }
}
```

```

        catch( MyExc1 u ) {
    }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 145  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ

- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
            + d.f(b)  
            + d.f(c) );  
    }  
}
```

- A. DDD

- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

13. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

---

14. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 111
- B. 010

- C. 110
  - D. 100
  - E. 101
- 

15. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;

- C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

16. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 4**

1. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
    }
}
```

```
catch( MyExc1 j ) {
    System.out.print(6);
    throw( new MyExc2() );
}
finally {
    System.out.print(7);
}
}
```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]){
    new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata private
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
                        + d.f(b)
                        + d.f(c));
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-

8. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3)  
    {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 111
- B. 010
- C. 110
- D. 100
- E. 101

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

A. C2

B. C3

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
    public static void main(String[] args){  
        new X().new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

A. X

B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
```

```

public static void main(String[] args)           }
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
}

A. C2
B. C3
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.



---


16. Qual è il risultato della compilazione ed esecuzione
del seguente programma?

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}
}

A. Z
B. XYZ
C. XXYZ
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 5**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
```

```
Boolean b2 = new Boolean(true);
b1 = new Boolean(true);
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(i3 == i1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b2 == b1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
```

- A. 111
- B. 010
- C. 110
- D. 100
- E. 101

3. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}

```

```

class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
- 

D. 423

E. Nessuna delle precedenti

---

9. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata private
- 

10. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
    }
}

```

```

C c = new C(); D d = new D();
System.out.println(d.f(a)
    + d.f(b)
    + d.f(c) );
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

**12. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

**15. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

```

```
public static void main(String[] args){  
    new X().new I().print();  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}
```

```
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 6**

1. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
- B. `d = b;`
- C. `c = d;`
- D. `b = c;`
- E. `b = (C1) d;`

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
```

```
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
```

- A. Errore a tempo di compilazione
- B. 145
- C. 145222222... (ciclo infinito)
- D. 14523Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

3. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 010

- C. 110
  - D. 100
  - E. 101
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}  
  
A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
```

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
        catch( MyExc1 u ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
  
    static void n() throws Exception {  
        try {  
            throw( new Exception() );  
        }
```

```
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

6. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z

- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

11. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata **private**

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c));  
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- 
13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 k ) {  
            System.out.print(5);  
        }  
    }  
}
```

```
        catch( MyExc1 j ) {  
            System.out.print(6);  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(7);  
        }  
    }  
}
```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
    }  
}
```

```
System.out.println( d.f(a)
+ d.f(b)
+ d.f(c) );  
}  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
```

```
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 7**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

5. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
```

```

public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
D. Un oggetto può non esistere dopo la sua dichiarazione  
E. Una classe non interna può essere dichiarata **private**
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }

    public static void main(String[] args) {
        new X().new I().print();
    }
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
        + d.f(b)
        + d.f(c));
}

```

```
    }  
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}
```

```
    }  
}
```

```
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

15. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 8**

1. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 010
- C. 110
- D. 100
- E. 101

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
```

```
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f();
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}
class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}
public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c) );
    }
}
```

- A. DDD
- B. ABC

- C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                           + d.f(b)
                           + d.f(c) );
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

8. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
    }  
}
```

```
}  
catch( MyExc1 u ) {  
}  
finally {  
    System.out.print(3);  
}  
}  
static void n() throws Exception {  
    try {  
        throw( new Exception() );  
    }  
    catch( MyExc3 e ) {  
        throw( new MyExc1() );  
    }  
    finally {  
        System.out.print(4);  
    }  
}  
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 k ) {  
            System.out.print(5);  
        }  
        catch( MyExc1 j ) {  
            System.out.print(6);  
            throw( new MyExc2() );  
        }  
    }  
}
```

```

        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
 B. XYZ  
 C. XXYZ  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {

```

```

            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 145  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.

- 
15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.

- E. Nessuna delle precedenti.
- 

16. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;  
D. f = (Exception []) n;  
E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 9

1. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata `private`

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ

- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`
- B. `d = b;`
- C. `c = d;`
- D. `b = c;`
- E. `b = (C1) d;`

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                q();  
                System.out.print(2);  
            }  
            catch( MyExc2 d ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc3() );  
            }  
        }  
        static void q() throws Exception {  
            try {
```

```

        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {

```

```

        String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();

```

```

        System.out.print("X");
    }

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {

```

```

        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 4213

- C. 41  
D. 423  
E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

```

- ```

public static void main(String[] args) {
    new X().new I().print();
}

```
- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
-

15. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 10**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
    }
}
```

```
catch( MyExc1 u ) {
}
finally {
    System.out.print(3);
}
}
static void n() throws Exception {
try {
    throw( new Exception() );
}
catch( MyExc3 e ) {
    throw( new MyExc1() );
}
finally {
    System.out.print(4);
}
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2

- B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
- B. XYZ
- C. XXYZ

- D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 
6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}
```

```

    }
}

static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
                        + d.f(b)
                        + d.f(c) );
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

12. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) { }
    Boolean b1;
    Boolean b2 = new Boolean(true);
    b1 = new Boolean(true);
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 111  
 B. 010  
 C. 110  
 D. 100  
 E. 101
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 145  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
 B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Un oggetto può non esistere dopo la sua dichiarazione  
 E. Una classe non interna può essere dichiarata **private**
- 

15. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 11**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}
```

A. 146732Exception in thread "main" MyExc3

B. 14666666... (ciclo infinito)

C. Errore a tempo di compilazione

D. 1463

E. Nessuna delle precedenti

---

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
```

```
}
```

A. C2

B. C3

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

6. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 111

B. 010

C. 110

D. 100

E. 101

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

- 
8. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

- 
9. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

- 
10. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata private

- 
11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
        catch( MyExc3 z ) {
```

```

        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X

- B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2

- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}
```

```
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 12**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
    }}
```

```
        System.out.print(3);
    }
}
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
        }
    }
}
```

```

        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

5. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
 B. f = (Exception []) z;  
 C. z = (Object [] []) f;  
 D. f = (Exception []) n;  
 E. Nessuno dei precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }

    public static void main(String[] args) {
        new X().new I().print();
    }
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

7. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        }
    }
}

```

```

        } else {
            System.out.print(0);
        }
    }

A. 111
B. 010
C. 110
D. 100
E. 101

```

---

```

        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {

```

- ```

        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```
- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {

```

```

Y(int i) {
    System.out.print("Y");
}
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {

```

```

        n();
        System.out.print(1);
    }
    catch( Exception w ) {
        System.out.print(2);
    }
    catch( MyExc3 z ) {
        throw( new Exception() );
    }
    catch( MyExc1 u ) {
    }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

A. Errore a tempo di compilazione

B. 4213

C. 41

D. 423

E. Nessuna delle precedenti

---

16. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 13**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X

- B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
```

```
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione  
 B. 145  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}
public static void main(String args[]) {
    new Z(1);
```

```
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```
}  
if(i3 == i1) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}  
if(b2 == b1) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}  
}  
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
}  
  
public static void main(String args[]) {  
    new I().print();  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {
```

```

Y(int i) {
    System.out.print("Y");
}
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)

```

```

throws Exception {
try {
    System.out.print(1);
    q();
    System.out.print(2);
}
catch( MyExc2 d ) {
    System.out.print(3);
}
finally {
    throw( new MyExc3() );
}
}
static void q() throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc3 k ) {
    System.out.print(5);
}
catch( MyExc1 j ) {
    System.out.print(6);
    throw( new MyExc2() );
}
finally {
    System.out.print(7);
}
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
                      + d.f(b)
                      + d.f(c));
}

```

```
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

15. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione

E. Una classe non interna può essere dichiarata **private**

---

16. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 14**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 145
- C. 145222222... (ciclo infinito)
- D. 14523Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
```

```

        n();
        System.out.print(1);
    }
    catch( Exception w ) {
        System.out.print(2);
    }
    catch( MyExc3 z ) {
        throw( new Exception() );
    }
    catch( MyExc1 u ) {
    }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }

    public static void main(String[] args) {
        new X().new I().print();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

8. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

11. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 010
- C. 110
- D. 100
- E. 101

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c));  
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println(d.f(a))  
    }  
}
```

```

        + d.f(b)
        + d.f(c) );
    }

A. DDD
B. ABC
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

```

+ d.f(b)
+ d.f(c) );
}

class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {

```

- 
- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 15**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
```

```
    void print() {
        System.out.println( X.s );
    }
}

public static void main(String args[]) {
    new I().print();
}

A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.
```

---

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-

4. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 111
- B. 010
- C. 110
- D. 100
- E. 101

5. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabili:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;

E. b = (C1) d;

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
        catch( MyExc1 u ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            throw( new Exception() );  
        }  
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}
```

```

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}

class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}

class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}

public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
}

```

```

    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

10. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}

class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}

```

```

class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
- B. 1466666... (ciclo infinito)

C. Errore a tempo di compilazione

D. 1463

E. Nessuna delle precedenti

---

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

A. DDD

B. ABC

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X  
B. Y  
C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
16. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 16

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X

- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}
```

```

class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c));
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
            + d.f(b)
            + d.f(c));
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
    }
}

```

```

void print() {
    System.out.println( X.s );
}

public static void main(String args[]) {
    new I().print();
}

A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }

    public static void main(String[] args) {
        new X().new I().print();
    }
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

```

```

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

```

static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {

```

```

        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]){
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

13. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {

```

```

        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3

- B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

16. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        }  
    }  
}
```

```
} else {  
    System.out.print(0);  
}  
if(i3 == i1) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}  
if(b2 == b1) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}  
}  
}  
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 17**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]){
    new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
                      + d.f(b)
                      + d.f(c));
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
```

```

        System.out.print(2);
    }
    catch( MyExc2 d ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc3() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z

- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
}  
  
public static void main(String[] args) {  
    new X().new I().print();  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```

        System.out.print(0);
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

```

        }
    finally {
        System.out.print(3);
    }
}
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

```
    }  
}  
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione

E. Una classe non interna può essere dichiarata **private**

---

16. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 18**

1. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
```

```
}  
class C extends B {
    String f(C c) {return "C";}
}  
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

A. X

B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

A. X

B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                n();  
                System.out.print(1);  
            }  
            catch( Exception w ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 z ) {  
                throw( new Exception() );  
            }  
            catch( MyExc1 u ) {  
            }  
            finally {  
                System.out.print(3);  
            }  
        }  
        static void n() throws Exception {  
            try {  
                throw( new Exception() );  
            }  
            catch( MyExc3 e ) {  
                throw( new MyExc1() );  
            }  
            finally {  
                System.out.print(4);  
            }  
        }  
    }
```

A. Errore a tempo di compilazione

B. 4213

C. 41

D. 423

E. Nessuna delle precedenti

- 
7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }
```

```

        }
    }

    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 145  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}

class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}

class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}

public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

10. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
 B. f = (Exception []) z;  
 C. z = (Object [] []) f;  
 D. f = (Exception []) n;  
 E. Nessuno dei precedenti
- 

11. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;  
 B. d = b;  
 C. c = d;  
 D. b = c;  
 E. b = (C1) d;

- 
12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 k ) {  
            System.out.print(5);  
        }  
        catch( MyExc1 j ) {  
            System.out.print(6);  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(7);  
        }  
    }  
}
```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi

- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata private
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}
```

```
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 19**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( x.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

3. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
        }
```

```

        q();
        System.out.print(2);
    }
    catch( MyExc2 d ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc3() );
    }
}

static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}

public static void main(String args[])
{

```

```

    new Z(1);
}
A. Z
B. XYZ
C. XXYZ
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
```

```
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
```

```

    }
}

class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

**12. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 4213

- C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

A. Z

```

- B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata private
- 

16. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 20**

1. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
    }
}
```

```
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

7. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;

E. Nessuno dei precedenti

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c) );  
    }  
}
```

A. DDD

B. ABC

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }
```

```
    }  
    catch( MyExc3 k ) {  
        System.out.print(5);  
    }  
    catch( MyExc1 j ) {  
        System.out.print(6);  
        throw( new MyExc2() );  
    }  
    finally {  
        System.out.print(7);  
    }  
}
```

A. 146732Exception in thread "main" MyExc3

B. 14666666... (ciclo infinito)

C. Errore a tempo di compilazione

D. 1463

E. Nessuna delle precedenti

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

A. X

B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }
```

```

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

13. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;

- B. d = b;  
 C. c = d;  
 D. b = c;  
 E. b = (C1) d;
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

15. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {

```

```
        System.out.print(0);
    }
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
```

```
class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}
public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 21**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 145
- C. 145222222... (ciclo infinito)
- D. 14523Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
```

```
throws Exception {
    try {
        System.out.print(1);
        q();
        System.out.print(2);
    }
    catch( MyExc2 d ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc3() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}
```

- A. 146732Exception in thread "main" MyExc3
- B. 146666666... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 1463
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
```

```

    }
}

public static void main(String args[]){
    new I().print();
}
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

```

    }
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

```

```

    }
}
```

```

public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
}

```

```

Y(int i) {
    System.out.print("Y");
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]){
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

11. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

12. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata private
- 

13. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
    public static void main(String[] args){
        new X().new I().print();
    }
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 22

1. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

2. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}

class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}

class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
```

```

public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 4213  
C. 41  
D. 423  
E. Nessuna delle precedenti
- 

#### 6. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata private
- 

#### 7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {

```

```

        System.out.print(2);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(3);
    }
}
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 
9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

- 
10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}
public static void main(String[] args) {
    new X().new I().print();
}
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

```

```

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 23

1. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata private

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ

C. XXYZ

- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;

C. `z = (Object [] []) f;`

D. `f = (Exception []) n;`

E. Nessuno dei precedenti

- 
5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";} }  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c)); } }
```

A. DDD

B. ABC

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s); } } }  
    public static void main(String[] args) {  
        new X().new I().print(); } }
```

A. X

B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
7. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`  
B. `d = b;`  
C. `c = d;`  
D. `b = c;`  
E. `b = (C1) d;`

- 
8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p(); }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() ); }  
        finally {  
            System.out.print(3); } }  
        static void p() {  
            try {  
                System.out.print(4);  
                throw( new Exception() ); }  
            catch( MyExc3 z ) {  
                throw( new MyExc2() ); }  
            finally {  
                System.out.print(5); } } } }
```

- A. Errore a tempo di compilazione  
B. 145

- 
- C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
}  
  
public static void main(String args[]) {  
    new I().print();  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 k ) {  
            System.out.print(5);  
        }  
        catch( MyExc1 j ) {  
            System.out.print(6);  
            throw( new MyExc2() );  
        }  
        finally {  
        }
```

```

        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
- 

13. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111  
 B. 010  
 C. 110  
 D. 100  
 E. 101
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {

```

```

public static void main(String [] argv)
throws Exception {
    try {
        n();
        System.out.print(1);
    }
    catch( Exception w ) {
        System.out.print(2);
    }
    catch( MyExc3 z ) {
        throw( new Exception() );
    }
    catch( MyExc1 u ) {
    }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {

```

```
        System.out.print("Z");
    }
    public static void main(String args[]){
        new Z(1);
    }
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
```

```
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 24**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
```

```
class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}
public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
```

```
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
}
```

```

    }

    public static void main(String args[]){
        new I().print();
    }
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}

class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}

class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}

public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}

class B extends A {
    String f(B b) {return "B";}
}

class C extends B {
    String f(C c) {return "C";}
}

class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}

public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
        + d.f(b)
        + d.f(c));
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. Errore a tempo di compilazione

B. 4213

C. 41

D. 423

E. Nessuna delle precedenti

---

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}
```

}

```
public static void main(String[] args) {
    new X().new I().print();
}
```

A. X

B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}
```

A. 146732Exception in thread "main" MyExc3

B. 14666666... (ciclo infinito)

C. Errore a tempo di compilazione

D. 1463

E. Nessuna delle precedenti

12. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 111
- B. 010
- C. 110
- D. 100
- E. 101

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] args) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {
```

```
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
        catch( MyExc3 z ) {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }
```

- A. Errore a tempo di compilazione
- B. 145
- C. 145222222... (ciclo infinito)
- D. 14523Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}
```

```
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

---

16. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 25**

1. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
```

```
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 
5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 
6. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
 B. Ogni valore in memoria non è associato ad un tipo di dato particolare

- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Un oggetto può non esistere dopo la sua dichiarazione  
 E. Una classe non interna può essere dichiarata `private`
- 
7. Quale output si ottiene invocando il metodo `p`?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111  
 B. 010  
 C. 110  
 D. 100  
 E. 101
- 
8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

```

```

    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

#### 11. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;

- C. `z = (Object [] []) f;`
  - D. `f = (Exception []) n;`
  - E. Nessuno dei precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
}
```

```
        System.out.print("Z");
    }
    public static void main(String args[]){
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
```

```
public void f(int i) throws Exception {
    System.out.println("C2");
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f();
    }
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 26**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}
```

```
} }
```

```
class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}  
  
A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
```

- 
5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 k ) {  
            System.out.print(5);  
        }  
        catch( MyExc1 j ) {  
            System.out.print(6);  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(7);  
        }  
    }  
}
```

- ```
    }  
}  
  
A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
```
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
}  
public static void main(String[] args)  
    throws Exception {  
    C1 c1 = new C2();  
    c1.f(1);  
}
```

A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";} }  
public static void main(String[] args) {  
    A a = new A(); B b = new B();  
    C c = new C(); D d = new D();  
}
```

```

        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c) );
    }
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z  
 B. XYZ  
 C. XXYZ  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {

```

```

            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

11. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
 B. f = (Exception []) z;  
 C. z = (Object [] []) f;  
 D. f = (Exception []) n;  
 E. Nessuno dei precedenti
-

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

```
finally {
    System.out.print(3);
}
}
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 145
- C. 145222222... (ciclo infinito)
- D. 14523Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

---

14. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata private

---

15. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
}

if(i3 == i1) {
    System.out.print(1);
} else {
    System.out.print(0);
}

if(b2 == b1) {
    System.out.print(1);
} else {
    System.out.print(0);
}

}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

16. Date le dichiarazioni:



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 27**

1. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata private

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
}
```

```

}
void m(String s3, Integer i1, Integer i3) {
    Boolean b1;
    Boolean b2 = new Boolean(true);
    b1 = new Boolean(true);
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

System.out.println(s);
}
}

public static void main(String[] args) {
    new X().new I().print();
}
}

A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
}

```

5. Date le dichiarazioni:

A. 111  
B. 010  
C. 110  
D. 100  
E. 101

---

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {

```

```

String f(D d) {return "D";}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
        + d.f(b)
        + d.f(c));
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

9. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;  
 B. d = b;  
 C. c = d;  
 D. b = c;  
 E. b = (C1) d;
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
}

```

```

        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
        catch( MyExc1 u ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            throw( new Exception() );  
        }  
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
}
```

```
}

public static void main(String args[]){
    new I().print();
}

}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
```

```
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}

public static void main(String args[]){
    new Z(1);
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 28**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
```

```
void print() {
    System.out.println( X.s );
}
public static void main(String args[]) {
    new I().print();
}
A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.
```

---

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

5. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;  
D. f = (Exception []) n;  
E. Nessuno dei precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

```

```

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}
class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
    public static void main(String[] args) {
        new X().new I().print();
    }
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
}

```

```

static void f(D d) {}
String f(D d) {return "D";}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
        + d.f(b)
        + d.f(c));
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
- B. 14666666... (ciclo infinito)

C. Errore a tempo di compilazione

D. 1463

E. Nessuna delle precedenti

---

#### 10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

A. Z
B. XYZ
C. XXYZ
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

#### 11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {

```

```

    }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

13. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;  
 B. d = b;  
 C. c = d;  
 D. b = c;  
 E. b = (C1) d;
- 

14. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

A. 111  
 B. 010  
 C. 110  
 D. 100  
 E. 101

---

15. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
 B. Ogni valore in memoria non è associato ad un tipo di dato particolare

- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}
```

```
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 29

1. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata private

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;
```

```

        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

6. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
    }
}

```

```

        finally {
            throw( new MyExc3() );
        }
    }
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}
}

```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

#### 10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

#### 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X

- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

---

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
}
```

```
X(int i) {
    this();
    System.out.print("X");
}
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]){
        new Z(1);
    }
}

A. Z
B. XYZ
C. XXYZ
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.
```

---

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
    public static void main(String[] args){
        new X().new I().print();
    }
}

A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.
```

---

15. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {
```

```
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
}  
public static void main(String[] args)  
    throws Exception {  
    C1 c1 = new C2();  
    c1.f(1);  
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 30**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
```

```
} catch( Exception w ) {
    System.out.print(2);
}
catch( MyExc3 z ) {
    throw( new Exception() );
}
catch( MyExc1 u ) {
}
finally {
    System.out.print(3);
}
}
static void n() throws Exception {
try {
    throw( new Exception() );
}
catch( MyExc3 e ) {
    throw( new MyExc1() );
}
finally {
    System.out.print(4);
}
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;

E. Nessuno dei precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";} }  
public static void main(String[] args) {  
    A a = new A(); B b = new B();  
    C c = new C(); D d = new D();  
    System.out.println(d.f(a)  
                      + d.f(b)  
                      + d.f(c)); } }
```

A. DDD

B. ABC

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";} }  
public static void main(String[] args) {  
    A a = new D(); B b = new D();  
    C c = new D(); D d = new D();  
    System.out.println( d.f(a)  
                      + d.f(b)  
                      + d.f(c)); } }
```

A. DDD

B. ABC

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    } }  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    } }  
public static void main(String[] args)  
throws Exception {  
    C1 c1 = new C2();  
    c1.f(1); }  
}  
  
A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
```

7. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    } }  
void m(String s3, Integer i1, Integer i3) {  
    Boolean b1;  
    Boolean b2 = new Boolean(true);  
    b1 = new Boolean(true);  
    if(s3 == s2) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    } }  
    if(i3 == i1) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    } }
```

```

    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 111  
 B. 010  
 C. 110  
 D. 100  
 E. 101
- 

8. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;  
 B. d = b;  
 C. c = d;  
 D. b = c;  
 E. b = (C1) d;
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 145  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

11. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
 B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri

- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata `private`

---

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 k ) {  
            System.out.print(5);  
        }  
    }  
}
```

```
        catch( MyExc1 j ) {  
            System.out.print(6);  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(7);  
        }  
    }  
}
```

- A. 146732Exception in thread "main" MyExc3
- B. 14666666... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 1463
- E. Nessuna delle precedenti

---

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 31

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata `private`

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c) );  
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                           + d.f(b)  
                           + d.f(c) );  
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.

- E. Nessuna delle precedenti.

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

8. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
        catch( MyExc3 z ) {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- ```
    }  
}  
  
A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
```
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

12. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {

```

```

public static void main(String [] argv)
    throws Exception {
    try {
        System.out.print(1);
        q();
        System.out.print(2);
    }
    catch( MyExc2 d ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc3() );
    }
}
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

16. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```
}  
if(i3 == i1) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}  
}  
if(b2 == b1) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}  
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 32

1. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata `private`

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
    throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 k ) {  
            System.out.print(5);  
        }  
        catch( MyExc1 j ) {  
            System.out.print(6);  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(7);  
        }  
    }  
}
```

- A. 146732Exception in thread "main" MyExc3
- B. 14666666... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 1463
- E. Nessuna delle precedenti

- 
4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
        catch( MyExc3 z ) {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c) );  
    }  
}
```

```
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

8. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {

```

```
            System.out.println( X.s );
        }
    }
}
```

```
public static void main(String args[]) {
    new I().print();
}
A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.
```

---

10. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

A. DDD

B. ABC

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

A. C2

B. C3

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

A. Z

B. XYZ

C. XXYZ

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

- 
15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
    }  
}
```

```

        catch( MyExc1 u ) {
    }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423

E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 33**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;

- D. b = c;
- E. b = (C1) d;

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[])
        throws Exception {
        new I().print();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata *private*

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }

```

```

public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a));
    }
}

```

```

        + d.f(b)
        + d.f(c) );
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

9. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;  
D. f = (Exception []) n;  
E. Nessuno dei precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
    }
}

```

```

        System.out.print("Z");
    }
    public static void main(String args[]){
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 010
- C. 110

D. 100

E. 101

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
}
```

```
}
```

```
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 34**

1. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
```

```
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
```

```

    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X
- B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

A. Errore a tempo di compilazione

B. 145

C. 145222222... (ciclo infinito)

D. 14523Exception in thread "main" MyExc3

E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
    }
}

```

```

        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi

- B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Un oggetto può non esistere dopo la sua dichiarazione  
 E. Una classe non interna può essere dichiarata **private**
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
    }
}

```

```

        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
    }
}

```

```

        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

15. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 35**

1. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
}
```

```
    finally {
        System.out.print(4);
    }
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
    public static void main(String[] args) {
        new X().new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
```

```

}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

#### 6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {

```

---

```

try {
    System.out.print(4);
    throw( new Exception() );
}
catch( MyExc3 z ) {
    throw( new MyExc2() );
}
finally {
    System.out.print(5);
}
}

```

- A. Errore a tempo di compilazione  
 B. 145  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

8. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111  
 B. 010  
 C. 110  
 D. 100  
 E. 101

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}
public static void main(String args[]) {
    new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD
- B. ABC

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

15. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 36**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
```

```
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata **private**

4. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 145
- C. 145222222... (ciclo infinito)
- D. 14523Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

---

6. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]){
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
```

```

public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }

    public static void main(String[] args) {
        new X().new I().print();
    }
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

11. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

A. 111  
B. 010  
C. 110  
D. 100  
E. 101

---

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
    }
}

```

```

}
catch( MyExc3 z ) {
    throw( new Exception() );
}
catch( MyExc1 u ) {
}
finally {
    System.out.print(3);
}
}

static void n() throws Exception {
try {
    throw( new Exception() );
}
catch( MyExc3 e ) {
    throw( new MyExc1() );
}
finally {
    System.out.print(4);
}
}
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
                      + d.f(b)
                      + d.f(c));
}

```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

```

```
public static void main(String args[]) {
    new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
```

```
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 37**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c));
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
    }
}
```

```

        System.out.print(3);
    }
}

static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

6. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;  
D. f = (Exception []) n;  
E. Nessuno dei precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

#### 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}

```

```

}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

12. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {

```

```

public void f() {
    System.out.println("C3");
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata **private**

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {
```

```
public void f(int i) throws Exception {  
    System.out.println("C2");  
}  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f();  
    }  
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 38**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata **private**

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
        + d.f(b)
        + d.f(c));
}
```

- A. DDD

- B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

6. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {

```

```

        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

7. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

```

```

    }

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

A. Z
B. XYZ
C. XXYZ
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 39**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X

- B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
```

```

    }
    catch( Exception w ) {
        System.out.print(2);
    }
    catch( MyExc3 z ) {
        throw( new Exception() );
    }
    catch( MyExc1 u ) {
    }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

#### 5. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        }
    }
}

```

```

    } else {
        System.out.print(0);
    }
}
A. 111
B. 010
C. 110
D. 100
E. 101

```

---

#### 6. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

#### 7. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
    }
}

```

```

    }
    catch( Exception u ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
    finally {
        System.out.print(3);
    }
}
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}
public static void main(String args[]) {
    new Z(1);
}

```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

12. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }

    public static void main(String[] args) {
        new X().new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
```

```
String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
}
String f(D d) {return "D";}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
        + d.f(b)
        + d.f(c));
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 146732Exception in thread "main" MyExc3

- B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {
```

```
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 40**

1. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}


- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

```

- 
5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {
```

```
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.

---

7. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;  
B. d = b;  
C. c = d;  
D. b = c;  
E. b = (C1) d;
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                n();  
                System.out.print(1);  
            }  
            catch( Exception w ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 z ) {  
                throw( new Exception() );  
            }  
            catch( MyExc1 u ) {  
            }  
            finally {  
                System.out.print(3);  
            }  
        }
```

```

static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

10. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
    }
}

```

```

m(s2, i2, new Integer(5));
}
void m(String s3, Integer i1, Integer i3) {
    Boolean b1;
    Boolean b2 = new Boolean(true);
    b1 = new Boolean(true);
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 111  
 B. 010  
 C. 110  
 D. 100  
 E. 101
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c));
    }
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.

- 
12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";} }  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        } catch( MyExc2 d ) {  
            System.out.print(3);  
        } finally {  
            throw( new MyExc3() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        } catch( MyExc3 k ) {
```

```
        System.out.print(5);  
    } catch( MyExc1 j ) {  
        System.out.print(6);  
        throw( new MyExc2() );  
    } finally {  
        System.out.print(7);  
    }  
}
```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        } catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        } finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        } catch( MyExc3 z ) {  
            throw( new MyExc2() );  
        } finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
-

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

- E. Nessuna delle precedenti.

- 
16. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata `private`
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 41**

1. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

2. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

6. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;

- B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

A. Z
B. XYZ
C. XXYZ
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

```

```

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]){
        new Z(1);
    }
}

```

- A. Z  
 B. XYZ  
 C. XXYZ  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}

```

```

public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
        + d.f(b)
        + d.f(c) );
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
    public static void main(String[] args) {
        new X().new I().print();
    }
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
}

```

```

static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione

- B. 4213  
C. 41  
D. 423  
E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
```

```
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}
```

- 
- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 42**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.
```

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
                      + d.f(b)
                      + d.f(c));
}
A. DDD
B. ABC
C. Errore a tempo di compilazione.
```

- D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

4. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
D. Un oggetto può non esistere dopo la sua dichiarazione  
E. Una classe non interna può essere dichiarata **private**
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

6. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;  
B. d = b;  
C. c = d;  
D. b = c;  
E. b = (C1) d;
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

8. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

**9. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

**10. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

A. Z
B. XYZ
C. XXYZ
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

**11. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
}

```

```

static void f() {}
String f(D d) {return "D";}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?
- 

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

16. Date le dichiarazioni:
- 

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 43**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
```

```
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
    }
}
```

```

        }
    finally {
        throw( new MyExc3() );
    }
}

static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}
}

```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

6. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
D. Un oggetto può non esistere dopo la sua dichiarazione  
E. Una classe non interna può essere dichiarata **private**
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
        }
    }
}

```

```

        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111  
B. 010  
C. 110  
D. 100  
E. 101
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

9. Quale output si ottiene invocando il metodo p?

- 
- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c));
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
    public static void main(String[] args) {
        new X().new I().print();
    }
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 44**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 14673Exception in thread "main" MyExc3
- B. 1466666... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 1463
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 145
- C. 145222222... (ciclo infinito)
- D. 14523Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}
```

```

    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
 B. XYZ  
 C. XXYZ  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}

```

```

class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {

```

```

            System.out.print(0);
        }
    }
}

A. 111
B. 010
C. 110
D. 100
E. 101

```

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;

- D. `f = (Exception []) n;`  
E. Nessuno dei precedenti

---

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
        catch( MyExc1 u ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            throw( new Exception() );  
        }  
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione  
B. 4213  
C. 41  
D. 423  
E. Nessuna delle precedenti

---

14. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `d = c;`

- B.  $d = b;$
- C.  $c = d;$
- D.  $b = c;$
- E.  $b = (C1) d;$

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X

- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata **private**



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 45**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
```

```

        this();
        System.out.print("X");
    }

}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
    }
}

```

```

void print() {
    System.out.println(s);
}

public static void main(String[] args) {
    new X().new I().print();
}

A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

8. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

if(i3 == i1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b2 == b1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}

```

- A. 111  
 B. 010  
 C. 110  
 D. 100  
 E. 101
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3

- B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

11. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
 B. f = (Exception []) z;  
 C. z = (Object [] []) f;  
 D. f = (Exception []) n;  
 E. Nessuno dei precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
- 

E. Errore a tempo di esecuzione.

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione

E. Una classe non interna può essere dichiarata private

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
}
```

```
}
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 46**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}
public static void main(String args[]){
    new Z(1);
}
```

A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.

3. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;

E. Nessuno dei precedenti

---

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";} }  
public static void main(String[] args) {  
    A a = new A(); B b = new B();  
    C c = new C(); D d = new D();  
    System.out.println(d.f(a)  
                      + d.f(b)  
                      + d.f(c)); }
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}  
  
A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
```

---

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
        catch( MyExc3 z ) {  
            throw( new MyExc2() );  
        }  
        finally {
```

```

        System.out.print(5);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 145  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {

```

```

        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 146666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

11. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;

```

```

Boolean b2 = new Boolean(true);
b1 = new Boolean(true);
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(i3 == i1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b2 == b1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z
- B. XYZ
- C. XXYZ

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

---

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
            + d.f(b)
            + d.f(c) );
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}

public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
- B. C3

- C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

16. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 47**

1. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

2. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
D. Un oggetto può non esistere dopo la sua dichiarazione  
E. Una classe non interna può essere dichiarata private
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}
public static void main(String args[]){
    new Z(1);
}

```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";} }  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c));  
    } }
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        } }  
    public static void main(String[] args) {  
        new X().new I().print();  
    } }
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc3() );  
        } }  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 k ) {  
            System.out.print(5);  
        }  
        catch( MyExc1 j ) {  
            System.out.print(6);  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(7);  
        } } }
```

- A. 146732Exception in thread "main" MyExc3

- B. 14666666... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 1463
- E. Nessuna delle precedenti

---

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

- E. Nessuna delle precedenti.

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
        catch( MyExc1 u ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            throw( new Exception() );  
        }  
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
}
```

```
        }
    public static void main(String args[]) {
        new I().print();
    }
}

A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.
```

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}
public static void main(String args[]) {
    new Z(1);
}
```

A. Z
B. XYZ
C. XXYZ
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 48**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
```

```

        System.out.print("Y");
    }

}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 4213

- C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

7. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

    }
    if(i3 == i1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

8. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabili:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
                q();
                System.out.print(2);
            }
            catch( MyExc2 d ) {

```

```

                System.out.print(3);
            }
            finally {
                throw( new MyExc3() );
            }
        }
        static void q() throws Exception {
            try {
                System.out.print(4);
                throw( new MyExc1() );
            }
            catch( MyExc3 k ) {
                System.out.print(5);
            }
            catch( MyExc1 j ) {
                System.out.print(6);
                throw( new MyExc2() );
            }
            finally {
                System.out.print(7);
            }
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

11. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata private
- 

12. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {  
        try {
```

```
        System.out.print(4);  
        throw( new Exception() );  
    }  
    catch( MyExc3 z ) {  
        throw( new MyExc2() );  
    }  
    finally {  
        System.out.print(5);  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}
```

```

class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 
16. Qual è il risultato della compilazione ed esecuzione del seguente programma?
- 

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 49**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c));
    }
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
    public static void main(String[] args) {
        new X().new I().print();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 146732Exception in thread "main" MyExc3
- B. 14666666... (ciclo infinito)
- C. Errore a tempo di compilazione

D. 1463

E. Nessuna delle precedenti

- 
4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

A. Z

B. XYZ

C. XXYZ

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

- 
5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

A. X

B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
6. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;  
D. f = (Exception []) n;  
E. Nessuno dei precedenti

- 
7. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
D. Un oggetto può non esistere dopo la sua dichiarazione  
E. Una classe non interna può essere dichiarata private

- 
8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
}
```

```

}
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 45  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

#### 10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

#### 11. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        }
    }
}

```

```

    } else {
        System.out.print(0);
    }
}

A. 111
B. 010
C. 110
D. 100
E. 101

}

System.out.print("Z");
}
public static void main(String args[]){
    new Z(1);
}
}

A. Z
B. XYZ
C. XXYZ
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

12. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
    }
}

```

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
}

A. C2
B. C3
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
}

```

```
C c = new D(); D d = new D();
System.out.println( d.f(a)
                    + d.f(b)
                    + d.f(c) );
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
```

```
public void f(int i) throws Exception {
    System.out.println("C2");
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f();
    }
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 50**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]){
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;

- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423

E. Nessuna delle precedenti

---

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

5. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata **private**

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}
```

```
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f();  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}  
  
A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
```

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}
```

```

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

9. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
- B. 010
- C. 110

- D. 100
  - E. 101
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
    }
}

```

```

}
catch( MyExc1 j ) {
    System.out.print(6);
    throw( new MyExc2() );
}
finally {
    System.out.print(7);
}
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
- 

**12. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 145  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

**15. Date le dichiarazioni:**

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 51**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
```

```
void print() {
    System.out.println(X.this.s);
}
}

public static void main(String args[]){
    new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args){
    new X().new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

#### 5. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;  
D. f = (Exception []) n;  
E. Nessuno dei precedenti
- 

#### 6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
}

```

```

    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

#### 7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
-

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";} }  
public static void main(String[] args) {  
    A a = new A(); B b = new B();  
    C c = new C(); D d = new D();  
    System.out.println(d.f(a)  
                      + d.f(b)  
                      + d.f(c) ); }
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
        catch( MyExc1 u ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            throw( new Exception() );  
        }  
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

11. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;

E. `b = (C1) d;`

---

12. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata `private`
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";} }  
public static void main(String[] args) {  
    A a = new D(); B b = new D();  
    C c = new D(); D d = new D();  
    System.out.println( d.f(a)  
                        + d.f(b)  
                        + d.f(c) ); } }
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Quale output si ottiene invocando il metodo `p`?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat(""); }  
void p() {  
    Integer i2;
```

```
    i2 = new Integer(5);  
    m(s2, i2, new Integer(5)); }  
void m(String s3, Integer i1, Integer i3) {  
    Boolean b1;  
    Boolean b2 = new Boolean(true);  
    b1 = new Boolean(true);  
    if(s3 == s2) {  
        System.out.print(1); } else {  
        System.out.print(0); }  
    if(i3 == i1) {  
        System.out.print(1); } else {  
        System.out.print(0); }  
    if(b2 == b1) {  
        System.out.print(1); } else {  
        System.out.print(0); } } }
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2); }  
        catch( MyExc2 d ) {  
            System.out.print(3); }  
        finally {  
            throw( new MyExc3() ); } }  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() ); }  
        catch( MyExc3 k ) {  
            System.out.print(5); } } }
```

```

    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}

```

```

}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 52**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
```

```
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

```
    }
}
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
```

```
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

8. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }

    public static void main(String[] args) {
        new X().new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
```

```
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
    }
}

```

- }
- }
- }
- }
- }
- }
- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
- 

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

**15. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
class Y extends X {
}

```

```

Y(int i) {
    System.out.print("Y");
}
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 53**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 145
- C. 145222222... (ciclo infinito)
- D. 14523Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

- 
2. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare

- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata private
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

---

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";} }  
public static void main(String[] args) {  
    A a = new A(); B b = new B();  
    C c = new C(); D d = new D();  
    System.out.println(d.f(a))
```

```
        + d.f(b)  
        + d.f(c) );  
    }  
}  
  
A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
```

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
        catch( MyExc1 u ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }
```

```

static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]){
    new I().print();
}
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {

```

```

        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
class Y extends X {

```

```

Y(int i) {
    System.out.print("Y");
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

14. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 54**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c));
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
    }
}
```

```
finally {
    throw( new MyExc3() );
}
}
static void q() throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc3 k ) {
    System.out.print(5);
}
catch( MyExc1 j ) {
    System.out.print(6);
    throw( new MyExc2() );
}
finally {
    System.out.print(7);
}
}
```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;

E. b = (C1) d;

- 
4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
        catch( MyExc1 u ) {}  
    }  
    finally {  
        System.out.print(3);  
    }  
}  
  
static void n() throws Exception {  
    try {  
        throw( new Exception() );  
    }  
    catch( MyExc3 e ) {  
        throw( new MyExc1() );  
    }  
}
```

```
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione  
B. 4213  
C. 41  
D. 423  
E. Nessuna delle precedenti
- 

6. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 111  
B. 010  
C. 110  
D. 100  
E. 101
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD
- B. ABC

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {

```

```

        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 145  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z  
 B. XYZ  
 C. XXYZ  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}

class C2 extends C1 {

```

```

    public void f(int i) {
        System.out.println("C2");
    }
}

class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}

public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
}

```

```
public static void main(String args[]) {
    new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione

E. Una classe non interna può essere dichiarata **private**

---

16. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 55**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
}
if(i3 == i1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b2 == b1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}
class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}
public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";} }  
public static void main(String[] args) {  
    A a = new A(); B b = new B();  
    C c = new C(); D d = new D();  
    System.out.println(d.f(a)  
        + d.f(b)  
        + d.f(c)); }
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata private
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";} }  
public static void main(String[] args) {  
    A a = new D(); B b = new D();  
    C c = new D(); D d = new D();  
    System.out.println( d.f(a)  
        + d.f(b)  
        + d.f(c)); }
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 k ) {  
            System.out.print(5);  
        }  
        catch( MyExc1 j ) {  
            System.out.print(6);  
            throw( new MyExc2() );  
        }  
    }  
}
```

```

    }
    finally {
        System.out.print(7);
    }
}
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
    public static void main(String[] args) {
        new X().new I().print();
    }
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]){
    new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
```

```
        System.out.print(3);
    }
}

static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 56

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata `private`

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
}
```

```

public static void main(String args[]) {
    new I().print();
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

5. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;  
D. f = (Exception []) n;  
E. Nessuno dei precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c));
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);

```

```

            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 146666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

**12. Date le dichiarazioni:**

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

**15. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
}

```

```

try {
    throw( new Exception() );
}
catch( MyExc3 e ) {
    throw( new MyExc1() );
}
finally {
    System.out.print(4);
}
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 57**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
D. Un oggetto può non esistere dopo la sua dichiarazione  
E. Una classe non interna può essere dichiarata **private**
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c));
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}

class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}

class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}

public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)

```

```

throws Exception {
    try {
        n();
        System.out.print(1);
    }
    catch( Exception w ) {
        System.out.print(2);
    }
    catch( MyExc3 z ) {
        throw( new Exception() );
    }
    catch( MyExc1 u ) {
    }
    finally {
        System.out.print(3);
    }
}

static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }

    static void q() throws Exception {
        try {

```

```

        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

12. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;  
D. f = (Exception []) n;  
E. Nessuno dei precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
}

```

```

Y(int i) {
    System.out.print("Y");
}
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]){
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Date le dichiarazioni:

---

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 58**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
    public static void main(String[] args) {
        new X().new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata **private**
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

```
} finally {
    System.out.print(3);
}
}
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
}
```

- A. Errore a tempo di compilazione
- B. 145
- C. 145222222... (ciclo infinito)
- D. 14523Exception in thread "main" MyExc3
- E. Nessuna delle precedenti
- 

7. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 010

- C. 110  
D. 100  
E. 101
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione  
B. 4213  
C. 41  
D. 423  
E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;  
B. d = b;  
C. c = d;  
D. b = c;  
E. b = (C1) d;
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
```

```

    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {

```

```

    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c));
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}

```

```

public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}

```

```

}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 59**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2

- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 146732Exception in thread "main" MyExc3
- B. 14666666... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 1463

E. Nessuna delle precedenti

- 
4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

A. DDD

B. ABC

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

- 
5. Dire quale delle seguenti affermazioni è vera:

A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi

B. Ogni valore in memoria non è associato ad un tipo di dato particolare

C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri

D. Un oggetto può non esistere dopo la sua dichiarazione

E. Una classe non interna può essere dichiarata **private**

- 
6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
}
```

```
}  
X(int i) {  
    this();  
    System.out.print("X");  
}  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

A. Z

B. XYZ

C. XXYZ

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

- 
7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

A. C2

B. C3

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";} }  
public static void main(String[] args) {  
    A a = new A(); B b = new B();  
    C c = new C(); D d = new D();  
    System.out.println(d.f(a)  
        + d.f(b)  
        + d.f(c)); } }
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- 
9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.s); } }  
    }  
  
    public static void main(String args[]) {  
        new I().print(); } }
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- 
10. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

- 
11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1); }  
        } catch( Exception w ) {  
            System.out.print(2); }  
        catch( MyExc3 z ) {  
            throw( new Exception() ); }  
        catch( MyExc1 u ) { }  
        finally {  
            System.out.print(3); } }  
static void n() throws Exception {  
    try {  
        throw( new Exception() ); }  
    catch( MyExc3 e ) {  
        throw( new MyExc1() ); }  
    finally {  
        System.out.print(4); } } }
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

---

12. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X

B. Y

- C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
        catch( MyExc3 z ) {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}
```

```
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 60**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c) );  
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
        catch( MyExc1 u ) {  
            System.out.print(3);  
        }  
    }  
}
```

```

    }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {

```

```

        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {

```

```

                System.out.print(1);
                q();
                System.out.print(2);
            }
            catch( MyExc2 d ) {
                System.out.print(3);
            }
            finally {
                throw( new MyExc3() );
            }
        }
static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

14. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD  
B. ABC

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

16. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 61

1. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata **private**

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";} }  
class B extends A {  
    String f(B b) {return "B";} }  
class C extends B {  
    String f(C c) {return "C";} }  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";} }  
public static void main(String[] args) {  
    A a = new A(); B b = new B();  
    C c = new C(); D d = new D();  
    System.out.println(d.f(a)  
        + d.f(b)  
        + d.f(c)); }
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
        catch( MyExc3 z ) {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 145
- C. 145222222... (ciclo infinito)
- D. 14523Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}
```

```

public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}
public static void main(String args[]) {
    new Z(1);
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
```

```
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}
```

```

    }

public static void main(String[] args){
    new X().new I().print();
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]){
    new I().print();
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

13. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

A. 111  

B. 010  

C. 110  

D. 100  

E. 101

```

---

14. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;  
B. d = b;  
C. c = d;  
D. b = c;  
E. b = (C1) d;
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}

```

```
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
```

```
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 62**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}
public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

- 
2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
}
```

```
public static void main(String args[]) {
    new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- 
3. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata *private*

- 
4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
- B. Y

- C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

6. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        }
    }
}
```

```
} else {
    System.out.print(0);
}
if(i3 == i1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(b2 == b1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
            + d.f(b)  
            + d.f(c));  
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println(d.f(a))  
    }  
}
```

```

        + d.f(b)
        + d.f(c) );
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
```

```
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}
```

- 
- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 63**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
```

```
System.out.println( d.f(a)
    + d.f(b)
    + d.f(c) );
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
    }
}
```

```

        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 145  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

#### 5. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111  
 B. 010  
 C. 110  
 D. 100  
 E. 101
- 

#### 6. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;  
 B. d = b;  
 C. c = d;  
 D. b = c;  
 E. b = (C1) d;
- 

#### 7. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
 B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Un oggetto può non esistere dopo la sua dichiarazione  
 E. Una classe non interna può essere dichiarata **private**
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
    }
}

```

```

        System.out.print(2);
    }
    catch( MyExc2 d ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc3() );
    }
}

static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
                      + d.f(b)
                      + d.f(c));
}

```

- A. DDD

- B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 4213  
C. 41  
D. 423  
E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

```

```

    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
 B. XYZ  
 C. XXYZ  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

13. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
 B. f = (Exception []) z;  
 C. z = (Object [] []) f;  
 D. f = (Exception []) n;  
 E. Nessuno dei precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

```

```
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 64**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
```

```
void print() {
    System.out.println(X.this.s);
}
}

public static void main(String args[]){
    new I().print();
}
}

A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.
```

---

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.

5. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
 B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Un oggetto può non esistere dopo la sua dichiarazione  
 E. Una classe non interna può essere dichiarata **private**

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
    }
}

```

```

        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c) );  
    }  
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Date le dichiarazioni:

```
class C0 extends Object {...}  
class C1 extends C0 {...}  
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;  
C0 c;  
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;

- D. b = c;
  - E. b = (C1) d;
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

---

12. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 111  
B. 010  
C. 110  
D. 100  
E. 101
- 

13. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;

- D. f = (Exception []) n;  
E. Nessuno dei precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
    public static void main(String args[]) {  
        new I().print();  
    }  
}  
A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
```

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
}  
public static void main(String[] args)  
throws Exception {  
    C1 c1 = new C2();  
    c1.f(1);  
}
```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

```

static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- 
- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 65**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
        }
    }
}
```

```
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}
```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

#### 6. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
D. Un oggetto può non esistere dopo la sua dichiarazione  
E. Una classe non interna può essere dichiarata private
- 

#### 7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
    public static void main(String[] args) {
        new X().new I().print();
    }
}
A. X
B. Y
C. Errore a tempo di compilazione.
D. Errore a tempo di esecuzione.
E. Nessuna delle precedenti.

```

---

#### 8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
}

```

```

String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c));
    }
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
 B. XYZ  
 C. XXYZ  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

11. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- B. 010
  - C. 110
  - D. 100
  - E. 101
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
}
```

```
X(int i) {
    this();
    System.out.print("X");
}
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 66**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( x.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2

- B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
```

```

C c = new D(); D d = new D();
System.out.println( d.f(a)
                    + d.f(b)
                    + d.f(c) );
}
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
- B. 14666666... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 1463
- E. Nessuna delle precedenti

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                            + d.f(b)
                            + d.f(c));
    }
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
    }
}

```

```

    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

12. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;  
 B. d = b;  
 C. c = d;  
 D. b = c;  
 E. b = (C1) d;
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {

```

```

        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
 B. XYZ  
 C. XXYZ  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
 B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Un oggetto può non esistere dopo la sua dichiarazione  
 E. Una classe non interna può essere dichiarata **private**
- 

15. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        }
    }
}

```

```
    } else {
        System.out.print(0);
    }
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
```

```
}  
  
class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}  
  
class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}  
public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 67**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
    }}
```

```
        System.out.print(3);
    }
}
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}
class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}
```

```

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z  
 B. XYZ  
 C. XXYZ  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}

class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}

class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }

    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

5. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;

- B. d = b;  
 C. c = d;  
 D. b = c;  
 E. b = (C1) d;
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z  
 B. XYZ  
 C. XXYZ  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
 B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Un oggetto può non esistere dopo la sua dichiarazione  
 E. Una classe non interna può essere dichiarata **private**
-

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 146732Exception in thread "main" MyExc3
- B. 14666666... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 1463
- E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
    }
}
```

```
}
```

```
catch( MyExc3 z ) {
    throw( new Exception() );
}
catch( MyExc1 u ) {
}
finally {
    System.out.print(3);
}
}

static void n() throws Exception {
try {
    throw( new Exception() );
}
catch( MyExc3 e ) {
    throw( new MyExc1() );
}
finally {
    System.out.print(4);
}
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41
- D. 423
- E. Nessuna delle precedenti

---

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

12. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

13. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]){
    new I().print();
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 
16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}

```

```

    }

class B extends A {
    String f(B b) {return "B";}
}

class C extends B {
    String f(C c) {return "C";}
}

class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 68**

1. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata **private**

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c));
    }
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X

- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {
```

```
Z(int i) {  
    super(i);  
    System.out.print("Z");  
}  
public static void main(String args[]) {  
    new Z(1);  
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

A. Z

B. XYZ

C. XXYZ

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                n();
                System.out.print(1);
            }
            catch( Exception w ) {
                System.out.print(2);
            }
            catch( MyExc3 z ) {
                throw( new Exception() );
            }
        }
}
```

```
    }
    catch( MyExc1 u ) {
    }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}
}
```

A. Errore a tempo di compilazione

B. 4213

C. 41

D. 423

E. Nessuna delle precedenti

11. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

A. z = (Object [] []) n;

B. f = (Exception []) z;

C. z = (Object [] []) f;

D. f = (Exception []) n;

E. Nessuno dei precedenti

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

```

}
finally {
    System.out.print(3);
}
}

static void p() {
try {
    System.out.print(4);
    throw( new Exception() );
}
catch( MyExc3 z ) {
    throw( new MyExc2() );
}
finally {
    System.out.print(5);
}
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

### 13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
    
```

- ```

                System.out.print(7);
            }
        }
    }

A. 146732Exception in thread "main" MyExc3
B. 146666666... (ciclo infinito)
C. Errore a tempo di compilazione
D. 1463
E. Nessuna delle precedenti

```
- 

### 14. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;  
B. d = b;  
C. c = d;  
D. b = c;  
E. b = (C1) d;
- 

### 15. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
    
```

```
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 111
- B. 010
- C. 110
- D. 100
- E. 101

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
```

```
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 69**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
```

```
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

6. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

7. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
                        + d.f(b)
                        + d.f(c) );
}

```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

9. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

A. 111

B. 010

C. 110

D. 100

E. 101

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] args)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
    }  
}
```

```
}  
finally {  
    throw( new MyExc3() );  
}  
}  
static void q() throws Exception {  
    try {  
        System.out.print(4);  
        throw( new MyExc1() );  
    }  
    catch( MyExc3 k ) {  
        System.out.print(5);  
    }  
    catch( MyExc1 j ) {  
        System.out.print(6);  
        throw( new MyExc2() );  
    }  
    finally {  
        System.out.print(7);  
    }  
}
```

A. 146732Exception in thread "main" MyExc3

B. 14666666... (ciclo infinito)

C. Errore a tempo di compilazione

D. 1463

E. Nessuna delle precedenti

---

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

A. C2

B. C3

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
}  
  
public static void main(String args[]) {  
    new I().print();  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- 
13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- 
14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}
```

```
}  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- 
15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
        catch( MyExc1 u ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            throw( new Exception() );  
        }  
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 4213
- C. 41

D. 423

E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
}
```

```
}
```

- ```
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
```
- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 70**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println( X.s );  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata `private`

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
    }
}

```

```

C c = new C(); D d = new D();
System.out.println(d.f(a)
    + d.f(b)
    + d.f(c) );
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

9. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;  
D. f = (Exception []) n;  
E. Nessuno dei precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                n();
                System.out.print(1);
            }
            catch( Exception w ) {
                System.out.print(2);
            }
            catch( MyExc3 z ) {
                throw( new Exception() );
            }
            catch( MyExc1 u ) {
            }
            finally {
                System.out.print(3);
            }
        }
}

```

```

static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
B. 4213  
C. 41  
D. 423  
E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
            + d.f(b)
            + d.f(c) );
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

12. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
}

```

```

}
void m(String s3, Integer i1, Integer i3) {
    Boolean b1;
    Boolean b2 = new Boolean(true);
    b1 = new Boolean(true);
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
    }
}

```

```

        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]){
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**15. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 71**

1. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}

class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}

class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}

public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.

- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}

class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}

class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}

public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
}

public static void main(String args[])
    throws Exception {
    I i = new I();
    i.print();
}
```

```
    new I().print();  
}  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

5. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c));  
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
    static void q() throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc3 k ) {  
            System.out.print(5);  
        }  
        catch( MyExc1 j ) {  
            System.out.print(6);  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(7);  
        }  
    }  
}
```

- A. 146732Exception in thread "main" MyExc3
- B. 14666666... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 1463
- E. Nessuna delle precedenti

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}
```

```

}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {

```

```

            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
  
    static void p() {  
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
        catch( MyExc3 z ) {  
            throw( new MyExc2() );  
        }  
    }  
}
```

```
}  
finally {  
    System.out.print(5);  
}  
}  
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare

- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

16. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
```

```
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 72**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
            + d.f(b)
            + d.f(c));
    }
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
```

```
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}
```

```
    }
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata private
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }

    public static void main(String[] args) {
        new X().new I().print();
    }
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 145
- C. 145222222... (ciclo infinito)
- D. 14523Exception in thread "main" MyExc3

E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
```

```
        System.out.print(7);
    }
}
}
A. 146732Exception in thread "main" MyExc3
B. 14666666... (ciclo infinito)
C. Errore a tempo di compilazione
D. 1463
E. Nessuna delle precedenti
```

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;

E. Nessuno dei precedenti

---

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

14. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
A. 111
B. 010
C. 110
D. 100
E. 101
```

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}
class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}
```

```
}

public static void main(String args[]) {
    new Z(1);
}

}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
```

```
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 73**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 145
- C. 145222222... (ciclo infinito)
- D. 14523Exception in thread "main" MyExc3
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
```

```

        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
 B. 14666666... (ciclo infinito)  
 C. Errore a tempo di compilazione  
 D. 1463  
 E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
 B. f = (Exception []) z;  
 C. z = (Object [] []) f;  
 D. f = (Exception []) n;  
 E. Nessuno dei precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
        + d.f(b)
        + d.f(c) );
}

```

- A. DDD  
 B. ABC  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
-

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c) );  
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

11. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata `private`

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            System.out.print(3);  
        }  
    }  
}
```

```

        throw( new Exception() );
    }
    catch( MyExc1 u ) {
    }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

13. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {
```

```
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 74**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
```

```
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c));  
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
}  
public static void main(String[] args)  
    throws Exception {  
    C1 c1 = new C2();  
    c1.f(1);  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

- 
4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

- E. Nessuna delle precedenti.
- 

6. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 111
- B. 010
- C. 110
- D. 100
- E. 101

- 
7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
            System.out.print(2);  
        }  
        catch( MyExc2 d ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc3() );  
        }  
    }  
}
```

```

    }
}

static void q() throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc3 k ) {
        System.out.print(5);
    }
    catch( MyExc1 j ) {
        System.out.print(6);
        throw( new MyExc2() );
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

8. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
D. Un oggetto può non esistere dopo la sua dichiarazione  
E. Una classe non interna può essere dichiarata **private**
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }

    public static void main(String[] args) {
        new X().new I().print();
    }
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

11. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;  
D. f = (Exception []) n;  
E. Nessuno dei precedenti
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 4213  
C. 41  
D. 423  
E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}

```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

---

16. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 75**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}
class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

```

        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
                      + d.f(b)
                      + d.f(c));
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
}
public static void main(String args[]) {
    new Z(1);
}

```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

7. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3  
B. 14666666... (ciclo infinito)  
C. Errore a tempo di compilazione  
D. 1463  
E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

10. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
B. f = (Exception []) z;  
C. z = (Object [] []) f;  
D. f = (Exception []) n;  
E. Nessuno dei precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
        catch( MyExc1 u ) {  
        }  
    finally {
```

```
        System.out.print(3);  
    }  
}  
static void n() throws Exception {  
    try {  
        throw( new Exception() );  
    }  
    catch( MyExc3 e ) {  
        throw( new MyExc1() );  
    }  
    finally {  
        System.out.print(4);  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 76**

1. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111
- B. 010
- C. 110
- D. 100
- E. 101

2. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
- B. d = b;
- C. c = d;
- D. b = c;
- E. b = (C1) d;

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
    public static void main(String args[]) {
        new I().print();
    }
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
        }
```

```

        System.out.print(1);
    }
    catch( Exception w ) {
        System.out.print(2);
    }
    catch( MyExc3 z ) {
        throw( new Exception() );
    }
    catch( MyExc1 u ) {
    }
    finally {
        System.out.print(3);
    }
}
static void n() throws Exception {
    try {
        throw( new Exception() );
    }
    catch( MyExc3 e ) {
        throw( new MyExc1() );
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
 B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Un oggetto può non esistere dopo la sua dichiarazione  
 E. Una classe non interna può essere dichiarata *private*
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

```

```

public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class B1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
        finally {
            System.out.print(3);
        }
    }
    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```

Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
-

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f(D d) {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new A(); B b = new B();  
        C c = new C(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c));  
    }  
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
- B. C3
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println(d.f(a)  
                           + d.f(b)  
                           + d.f(c));  
    }  
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {
```

```
Y(int i) {  
    System.out.print("Y");  
}  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
}  
public static void main(String args[]) {  
    new Z(1);  
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 77**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 146732Exception in thread "main" MyExc3
- B. 14666666... (ciclo infinito)
- C. Errore a tempo di compilazione
- D. 1463
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.

3. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

        System.out.print(0);
    }
    if(b2 == b1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 111  
 B. 010  
 C. 110  
 D. 100  
 E. 101
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
    public static void main(String[] args) {
        new X().new I().print();
    }
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

5. Date le dichiarazioni:

```

class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}

```

e le dichiarazioni di variabile:

```

C1 b;
C0 c;
Sub2 d;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;

- B. d = b;  
 C. c = d;  
 D. b = c;  
 E. b = (C1) d;
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
 B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Un oggetto può non esistere dopo la sua dichiarazione  
 E. Una classe non interna può essere dichiarata **private**
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
}

public static void main(String args[]) {
    new Z(1);
}

```

- A. Z  
 B. XYZ  
 C. XXYZ  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
-

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) throws Exception {  
        System.out.println("C2");  
    }  
}  
class C3 extends C1 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
  
    public static void main(String args[]) {  
        new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            n();  
            System.out.print(1);  
        }  
        catch( Exception w ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 z ) {  
            throw( new Exception() );  
        }  
        catch( MyExc1 u ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            throw( new Exception() );  
        }  
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
}
```

```

        }
    }

    static void p() {
        try {
            System.out.print(4);
            throw( new Exception() );
        }
        catch( MyExc3 z ) {
            throw( new MyExc2() );
        }
        finally {
            System.out.print(5);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 145  
C. 145222222... (ciclo infinito)  
D. 14523Exception in thread "main" MyExc3  
E. Nessuna delle precedenti
- 
12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }

    public static void main(String args[]) {
        new I().print();
    }
}

```

- A. X  
B. Y  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {

```

```

        String f(C c) {return "C";}
}

class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                           + d.f(b)
                           + d.f(c));
    }
}

```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

15. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {
```

```
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 78**

1. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
  - B. f = (Exception []) z;
  - C. z = (Object [] []) f;
  - D. f = (Exception []) n;
  - E. Nessuno dei precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
    }
}
```

```
catch( MyExc1 j ) {
    System.out.print(6);
    throw( new MyExc2() );
}
finally {
    System.out.print(7);
}
}
```

- A. 146732Exception in thread "main" MyExc3
  - B. 14666666... (ciclo infinito)
  - C. Errore a tempo di compilazione
  - D. 1463
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;
  - B. d = b;
  - C. c = d;
  - D. b = c;
  - E. b = (C1) d;
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
}
```

```

    }

public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 4213
  - C. 41
  - D. 423
  - E. Nessuna delle precedenti
- 

#### 6. Quale output si ottiene invocando il metodo p?

```

class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 111
  - B. 010
  - C. 110
  - D. 100
  - E. 101
- 

#### 7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
}

```

```
    c1.f(1);  
}  
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z
  - B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    private static String s = "X";  
    class I {  
        String s = "Y";  
        void print() {  
            System.out.println(s);  
        }  
    }
```

```
    }  
  
    public static void main(String[] args) {  
        new X().new I().print();  
    }  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
        catch( MyExc3 z ) {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 145
  - C. 145222222... (ciclo infinito)
  - D. 14523Exception in thread "main" MyExc3
  - E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
    public static void main(String[] args)
        throws Exception {
        C1 c1 = new C2();
        c1.f(1);
    }
}

```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
        + d.f(b)
        + d.f(c));
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new D(); B b = new D();
    C c = new D(); D d = new D();
    System.out.println( d.f(a)
        + d.f(b)
        + d.f(c));
}

```

- A. DDD
  - B. ABC
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}
public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}

class Z extends Y {
    Z(int i) {
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}

```

A. Z

- B. XYZ
  - C. XXYZ
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
  - B. Ogni valore in memoria non è associato ad un tipo di dato particolare
  - C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
  - D. Un oggetto può non esistere dopo la sua dichiarazione
  - E. Una classe non interna può essere dichiarata **private**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 79**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}

public static void main(String[] args) {
    new X().new I().print();
}
```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
}
public static void main(String[] args) {
    A a = new A(); B b = new B();
    C c = new C(); D d = new D();
    System.out.println(d.f(a)
                      + d.f(b)
                      + d.f(c));
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 146732Exception in thread "main" MyExc3
- B. 14666666... (ciclo infinito)
- C. Errore a tempo di compilazione

D. 1463

E. Nessuna delle precedenti

- 
4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        super(i);  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}
```

```
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {  
    String f(A a) {return "A";}  
}  
class B extends A {  
    String f(B b) {return "B";}  
}  
class C extends B {  
    String f(C c) {return "C";}  
}  
class D extends C {  
    static void f() {}  
    String f(D d) {return "D";}  
    public static void main(String[] args) {  
        A a = new D(); B b = new D();  
        C c = new D(); D d = new D();  
        System.out.println( d.f(a)  
                            + d.f(b)  
                            + d.f(c) );  
    }  
}
```

- A. DDD  
B. ABC  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
    }  
}
```

```

        catch( Exception u ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    finally {
        System.out.print(3);
    }
}
static void p() {
    try {
        System.out.print(4);
        throw( new Exception() );
    }
    catch( MyExc3 z ) {
        throw( new MyExc2() );
    }
    finally {
        System.out.print(5);
    }
}
}

```

- A. Errore a tempo di compilazione  
 B. 145  
 C. 145222222... (ciclo infinito)  
 D. 14523Exception in thread "main" MyExc3  
 E. Nessuna delle precedenti
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) {
        System.out.println("C2");
    }
}
class C3 extends C2 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}

```

- A. C2  
 B. C3  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}
public static void main(String args[]) {
    new I().print();
}

```

- A. X  
 B. Y  
 C. Errore a tempo di compilazione.  
 D. Errore a tempo di esecuzione.  
 E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            n();
            System.out.print(1);
        }
        catch( Exception w ) {
            System.out.print(2);
        }
        catch( MyExc3 z ) {
            throw( new Exception() );
        }
        catch( MyExc1 u ) {
        }
        finally {
            System.out.print(3);
        }
    }
    static void n() throws Exception {
        try {
            throw( new Exception() );
        }
        catch( MyExc3 e ) {
            throw( new MyExc1() );
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 4213  
 C. 41  
 D. 423  
 E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```
class C {
    private String s1 = "";
    private String s2 = s1.concat("");
    void p() {
        Integer i2;
        i2 = new Integer(5);
        m(s2, i2, new Integer(5));
    }
    void m(String s3, Integer i1, Integer i3) {
        Boolean b1;
        Boolean b2 = new Boolean(true);
        b1 = new Boolean(true);
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(b2 == b1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 111  
 B. 010  
 C. 110  
 D. 100  
 E. 101
- 

12. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. d = c;  
 B. d = b;  
 C. c = d;  
 D. b = c;  
 E. b = (C1) d;
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi  
 B. Ogni valore in memoria non è associato ad un tipo di dato particolare  
 C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri  
 D. Un oggetto può non esistere dopo la sua dichiarazione  
 E. Una classe non interna può essere dichiarata **private**
- 

14. Date le dichiarazioni:

```
Exception [] f;
Object [] n;
Object [] [] z;
n = new Object [1] [9];
f = new Exception [9];
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;  
 B. f = (Exception []) z;  
 C. z = (Object [] []) f;  
 D. f = (Exception []) n;  
 E. Nessuno dei precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
```

```
public void f() {  
    System.out.println("C3");  
}  
public static void main(String[] args)  
    throws Exception {  
    C1 c1 = new C2();  
    c1.f(1);  
}
```

- A. C2
  - B. C3
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```
class X {  
    String s = "X";  
    static class I {  
        String s = "Y";  
        void print() {  
            System.out.println(X.this.s);  
        }  
    }  
}  
  
public static void main(String args[]) {  
    new I().print();  
}
```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 80**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f() {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new D(); B b = new D();
        C c = new D(); D d = new D();
        System.out.println( d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    X() {
        System.out.print("X");
    }
    X(int i) {
        this();
        System.out.print("X");
    }
}

class Y extends X {
    Y(int i) {
        System.out.print("Y");
    }
}
```

```
} }
```

```
class Z extends Y {
    Z(int i) {
        super(i);
        System.out.print("Z");
    }
    public static void main(String args[]) {
        new Z(1);
    }
}
```

- A. Z
- B. XYZ
- C. XXYZ
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class A {
    String f(A a) {return "A";}
}
class B extends A {
    String f(B b) {return "B";}
}
class C extends B {
    String f(C c) {return "C";}
}
class D extends C {
    static void f(D d) {}
    String f(D d) {return "D";}
    public static void main(String[] args) {
        A a = new A(); B b = new B();
        C c = new C(); D d = new D();
        System.out.println(d.f(a)
                            + d.f(b)
                            + d.f(c) );
    }
}
```

- A. DDD
- B. ABC
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
            System.out.print(2);
        }
        catch( MyExc2 d ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc3() );
        }
    }
    static void q() throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc3 k ) {
            System.out.print(5);
        }
        catch( MyExc1 j ) {
            System.out.print(6);
            throw( new MyExc2() );
        }
        finally {
            System.out.print(7);
        }
    }
}
```

A. 146732Exception in thread "main" MyExc3

B. 14666666... (ciclo infinito)

C. Errore a tempo di compilazione

D. 1463

E. Nessuna delle precedenti

---

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {
    private static String s = "X";
    class I {
        String s = "Y";
        void print() {
            System.out.println(s);
        }
    }
}
```

}

```
public static void main(String[] args) {
    new X().new I().print();
}
```

}

A. X

B. Y

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

6. Date le dichiarazioni:

```
class C0 extends Object {...}
class C1 extends C0 {...}
class Sub2 extends C0 {...}
```

e le dichiarazioni di variabile:

```
C1 b;
C0 c;
Sub2 d;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

A. d = c;

B. d = b;

C. c = d;

D. b = c;

E. b = (C1) d;

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {
    abstract void f(int i) throws Exception;
}
class C2 extends C1 {
    public void f(int i) throws Exception {
        System.out.println("C2");
    }
}
class C3 extends C1 {
    public void f() {
        System.out.println("C3");
    }
}
public static void main(String[] args)
    throws Exception {
    C1 c1 = new C2();
    c1.f(1);
}
```

A. C2

B. C3

C. Errore a tempo di compilazione.

D. Errore a tempo di esecuzione.

E. Nessuna delle precedenti.

---

8. Dire quale delle seguenti affermazioni è vera:

- A. La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- B. Ogni valore in memoria non è associato ad un tipo di dato particolare
- C. Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D. Un oggetto può non esistere dopo la sua dichiarazione
- E. Una classe non interna può essere dichiarata **private**

---

9. Quale output si ottiene invocando il metodo p?

```
class C {  
    private String s1 = "";  
    private String s2 = s1.concat("");  
    void p() {  
        Integer i2;  
        i2 = new Integer(5);  
        m(s2, i2, new Integer(5));  
    }  
    void m(String s3, Integer i1, Integer i3) {  
        Boolean b1;  
        Boolean b2 = new Boolean(true);  
        b1 = new Boolean(true);  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(b2 == b1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

A. 111

B. 010

C. 110

D. 100

E. 101

---

10. Date le dichiarazioni:

```
Exception [] f;  
Object [] n;  
Object [] [] z;  
n = new Object [1] [9];  
f = new Exception [9];  
z = new Object [8] [5];
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. z = (Object [] []) n;
- B. f = (Exception []) z;
- C. z = (Object [] []) f;
- D. f = (Exception []) n;
- E. Nessuno dei precedenti

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class B1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception u ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(4);  
            throw( new Exception() );  
        }  
        catch( MyExc3 z ) {  
            throw( new MyExc2() );  
        }  
        finally {  
            System.out.print(5);  
        }  
    }  
}
```

A. Errore a tempo di compilazione

B. 145

C. 145222222... (ciclo infinito)

D. 14523Exception in thread "main" MyExc3

E. Nessuna delle precedenti

- 
12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
class X {  
    X() {  
        System.out.print("X");  
    }  
    X(int i) {  
        this();  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    Y(int i) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    Z(int i) {  
        System.out.print("Z");  
    }  
    public static void main(String args[]) {  
        new Z(1);  
    }  
}
```

- A. Z  
B. XYZ  
C. XXYZ  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                n();  
                System.out.print(1);  
            }  
            catch( Exception w ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 z ) {  
                throw( new Exception() );  
            }  
            catch( MyExc1 u ) {  
            }  
        }
```

```
        finally {  
            System.out.print(3);  
        }  
    }  
    static void n() throws Exception {  
        try {  
            throw( new Exception() );  
        }  
        catch( MyExc3 e ) {  
            throw( new MyExc1() );  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione  
B. 4213  
C. 41  
D. 423  
E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
abstract class C1 {  
    abstract void f(int i) throws Exception;  
}  
class C2 extends C1 {  
    public void f(int i) {  
        System.out.println("C2");  
    }  
}  
class C3 extends C2 {  
    public void f() {  
        System.out.println("C3");  
    }  
    public static void main(String[] args)  
        throws Exception {  
        C1 c1 = new C2();  
        c1.f(1);  
    }  
}
```

- A. C2  
B. C3  
C. Errore a tempo di compilazione.  
D. Errore a tempo di esecuzione.  
E. Nessuna delle precedenti.
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

class X {
    String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println(X.this.s);
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
  - B. Y
  - C. Errore a tempo di compilazione.
  - D. Errore a tempo di esecuzione.
  - E. Nessuna delle precedenti.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

---

```

class X {
    private static String s = "X";
    static class I {
        String s = "Y";
        void print() {
            System.out.println( X.s );
        }
    }
}

public static void main(String args[]) {
    new I().print();
}

```

- A. X
- B. Y
- C. Errore a tempo di compilazione.
- D. Errore a tempo di esecuzione.
- E. Nessuna delle precedenti.

**Prova n. 1**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 2**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 3**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 4**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 5**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 6**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 8**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 9**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 10**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 11**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 12**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 13**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 15**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 16**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

24 luglio 2017

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 1:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 2:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  |   |   | ■ |   |   |
| 2  |   |   |   |   | ■ |
| 3  | ■ |   |   |   |   |
| 4  |   | ■ |   |   |   |
| 5  |   |   | ■ |   |   |
| 6  |   |   | ■ |   |   |
| 7  |   |   |   | ■ |   |
| 8  |   |   |   | ■ |   |
| 9  | ■ |   |   |   |   |
| 10 |   |   |   |   | ■ |
| 11 | ■ |   |   |   |   |
| 12 |   |   |   | ■ |   |
| 13 | ■ |   |   |   |   |
| 14 |   |   |   |   | ■ |
| 15 |   |   |   | ■ |   |
| 16 |   |   |   |   | ■ |

Risultato prova n. 3:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   |   |   |   | ■ |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  |   |   |   |   | ■ | ■ |   |
| 6  |   |   | ■ | ■ |   |   |   |
| 7  |   |   |   | ■ | ■ |   |   |
| 8  |   |   |   |   | ■ | ■ |   |
| 9  |   |   |   |   | ■ | ■ |   |
| 10 |   |   | ■ | ■ |   |   |   |
| 11 | ■ | ■ |   |   |   |   |   |
| 12 | ■ | ■ |   |   |   |   |   |
| 13 |   |   |   | ■ | ■ |   |   |
| 14 |   |   |   |   | ■ | ■ |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 |   | ■ | ■ |   |   |   |   |

Risultato prova n. 4:

|    |   |   |   |   |   |   |
|----|---|---|---|---|---|---|
| 1  |   |   |   |   |   |   |
| 2  |   |   |   |   |   |   |
| 3  |   | ■ |   |   |   |   |
| 4  |   |   | ■ |   |   |   |
| 5  |   | ■ |   |   |   |   |
| 6  |   |   |   |   |   | ■ |
| 7  |   |   |   | ■ |   |   |
| 8  |   | ■ |   |   |   |   |
| 9  |   |   |   |   | ■ |   |
| 10 |   |   |   | ■ |   |   |
| 11 |   |   |   | ■ |   |   |
| 12 | ■ | ■ |   |   |   |   |
| 13 | ■ | ■ |   |   |   |   |
| 14 |   |   | ■ |   |   |   |
| 15 |   |   | ■ | ■ |   |   |
| 16 |   |   | ■ | ■ |   |   |

Risultato prova n. 5:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 6:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 7:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 8:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 9:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 10:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   | ■ | ■ |   |   |
| 3  |  |   |   | ■ | ■ |   |   |
| 4  |  |   |   |   |   |   | ■ |
| 5  |  | ■ | ■ |   |   |   |   |
| 6  |  |   |   |   | ■ | ■ |   |
| 7  |  |   |   |   | ■ | ■ |   |
| 8  |  | ■ | ■ |   |   |   |   |
| 9  |  |   |   | ■ | ■ |   |   |
| 10 |  |   |   |   | ■ | ■ |   |
| 11 |  | ■ | ■ |   |   |   |   |
| 12 |  |   | ■ | ■ |   |   |   |
| 13 |  | ■ | ■ |   |   |   |   |
| 14 |  |   | ■ | ■ |   |   |   |
| 15 |  |   |   | ■ | ■ |   |   |
| 16 |  |   | ■ | ■ |   |   |   |

Risultato prova n. 11:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 12:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  |   | ■ |   |   |   |   |
| 2  |  |   |   | ■ |   |   |   |
| 3  |  |   |   |   | ■ |   |   |
| 4  |  |   |   |   |   | ■ |   |
| 5  |  | ■ |   |   |   |   |   |
| 6  |  |   | ■ |   |   |   |   |
| 7  |  |   |   | ■ |   |   |   |
| 8  |  |   |   |   | ■ |   |   |
| 9  |  |   |   |   |   | ■ |   |
| 10 |  | ■ |   |   |   |   |   |
| 11 |  |   |   |   |   | ■ |   |
| 12 |  | ■ |   |   |   |   |   |
| 13 |  |   |   |   |   |   | ■ |
| 14 |  |   |   | ■ |   |   |   |
| 15 |  |   |   |   | ■ |   |   |
| 16 |  | ■ |   |   |   |   |   |

Risultato prova n. 13:

|    |   |   |   |   |   |   |    |
|----|---|---|---|---|---|---|----|
| 1  |   |   |   |   |   |   | 1  |
| 2  |   | 1 |   |   |   |   | 2  |
| 3  |   | 1 |   |   |   |   | 3  |
| 4  |   |   | 1 |   |   |   | 4  |
| 5  |   | 1 |   |   |   |   | 5  |
| 6  |   |   | 1 |   |   |   | 6  |
| 7  |   | 1 |   |   |   |   | 7  |
| 8  |   |   |   |   | 1 |   | 8  |
| 9  |   |   |   | 1 | 1 |   | 9  |
| 10 |   |   |   | 1 | 1 |   | 10 |
| 11 |   |   |   |   | 1 | 1 | 11 |
| 12 |   |   |   |   | 1 | 1 | 12 |
| 13 |   |   |   | 1 | 1 |   | 13 |
| 14 |   |   |   | 1 | 1 |   | 14 |
| 15 |   |   | 1 | 1 |   |   | 15 |
| 16 | 1 | 1 |   |   |   |   | 16 |

Risultato prova n. 14:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 15:

|    |  |   |   |   |   |
|----|--|---|---|---|---|
| 1  |  |   |   |   |   |
| 2  |  |   |   |   |   |
| 3  |  | ■ |   |   |   |
| 4  |  |   |   | ■ |   |
| 5  |  |   | ■ |   |   |
| 6  |  | ■ |   |   |   |
| 7  |  |   | ■ |   |   |
| 8  |  |   |   |   | ■ |
| 9  |  |   |   | ■ |   |
| 10 |  | ■ |   |   |   |
| 11 |  | ■ |   |   |   |
| 12 |  |   | ■ |   |   |
| 13 |  |   |   | ■ |   |
| 14 |  | ■ |   |   |   |
| 15 |  |   |   |   | ■ |
| 16 |  |   |   | ■ |   |

Risultato prova n. 16:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 17:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 18:

|    |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|--|
| 1  |   | ■ |   |   |   |   |  |
| 2  |   |   |   | ■ |   |   |  |
| 3  |   |   |   |   | ■ |   |  |
| 4  |   |   |   |   |   | ■ |  |
| 5  |   |   |   |   |   | ■ |  |
| 6  |   |   |   | ■ |   |   |  |
| 7  |   |   | ■ |   |   |   |  |
| 8  |   |   |   | ■ |   |   |  |
| 9  |   | ■ |   |   |   |   |  |
| 10 |   |   | ■ |   |   |   |  |
| 11 |   | ■ |   |   |   |   |  |
| 12 |   | ■ |   |   |   |   |  |
| 13 |   |   |   | ■ |   |   |  |
| 14 |   |   | ■ |   |   |   |  |
| 15 |   |   |   |   | ■ |   |  |
| 16 | ■ |   |   |   |   |   |  |

Risultato prova n. 19:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 20:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   |   |   |   | ■ |
| 3  |  |   |   | ■ | ■ |   |   |
| 4  |  |   |   |   |   | ■ | ■ |
| 5  |  |   |   |   | ■ | ■ |   |
| 6  |  | ■ | ■ |   |   |   |   |
| 7  |  |   | ■ | ■ |   |   |   |
| 8  |  | ■ | ■ |   |   |   |   |
| 9  |  |   |   | ■ | ■ |   |   |
| 10 |  |   |   |   | ■ | ■ |   |
| 11 |  | ■ | ■ |   |   |   |   |
| 12 |  |   |   |   | ■ | ■ |   |
| 13 |  |   |   | ■ | ■ |   |   |
| 14 |  | ■ | ■ |   |   |   |   |
| 15 |  |   | ■ | ■ |   |   |   |
| 16 |  |   | ■ | ■ |   |   |   |

Risultato prova n. 21:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 22:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 23:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 24:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 25:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  |   |   |   |   |   | ■ | ■ |
| 6  | ■ | ■ |   |   |   |   |   |
| 7  |   |   |   | ■ | ■ |   |   |
| 8  |   |   | ■ | ■ |   |   |   |
| 9  |   |   |   | ■ | ■ |   |   |
| 10 |   |   | ■ | ■ |   |   |   |
| 11 | ■ | ■ |   |   |   |   |   |
| 12 | ■ | ■ |   |   |   |   |   |
| 13 | ■ | ■ |   |   |   |   |   |
| 14 |   |   |   |   | ■ | ■ |   |
| 15 |   |   |   |   | ■ | ■ |   |
| 16 |   |   |   | ■ |   |   |   |

Risultato prova n. 26:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 27:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 28:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 29:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  | ■ | ■ |   |   |   |   |   |
| 3  | ■ | ■ |   |   |   |   |   |
| 4  |   |   |   | ■ | ■ |   |   |
| 5  |   |   | ■ | ■ |   |   |   |
| 6  | ■ | ■ |   |   |   |   |   |
| 7  |   |   |   |   | ■ | ■ |   |
| 8  |   |   |   | ■ | ■ |   |   |
| 9  |   |   |   | ■ | ■ |   |   |
| 10 | ■ | ■ |   |   |   |   |   |
| 11 |   |   |   |   | ■ | ■ |   |
| 12 |   |   | ■ | ■ |   |   |   |
| 13 |   |   |   |   |   | ■ | ■ |
| 14 |   |   |   |   | ■ | ■ |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 |   |   |   | ■ | ■ |   |   |

Risultato prova n. 30:

|    |   |  |   |   |   |   |  |
|----|---|--|---|---|---|---|--|
| 1  |   |  | ■ |   |   |   |  |
| 2  |   |  |   | ■ |   |   |  |
| 3  |   |  |   |   | ■ |   |  |
| 4  | ■ |  |   |   |   |   |  |
| 5  |   |  |   | ■ |   |   |  |
| 6  |   |  | ■ |   |   |   |  |
| 7  |   |  |   |   | ■ |   |  |
| 8  | ■ |  |   |   |   |   |  |
| 9  |   |  |   | ■ |   |   |  |
| 10 | ■ |  |   |   |   |   |  |
| 11 |   |  | ■ |   |   |   |  |
| 12 |   |  |   | ■ |   |   |  |
| 13 | ■ |  |   |   |   |   |  |
| 14 | ■ |  |   |   |   | ■ |  |
| 15 | ■ |  |   |   |   |   |  |
| 16 |   |  |   |   | ■ |   |  |

Risultato prova n. 31:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 32:

|    |  |   |   |   |   |
|----|--|---|---|---|---|
| 1  |  |   |   |   |   |
| 2  |  |   |   |   |   |
| 3  |  | ■ |   |   |   |
| 4  |  |   |   |   | ■ |
| 5  |  |   | ■ |   |   |
| 6  |  |   |   | ■ |   |
| 7  |  |   |   |   | ■ |
| 8  |  |   |   | ■ |   |
| 9  |  | ■ |   |   |   |
| 10 |  |   |   |   | ■ |
| 11 |  | ■ |   |   |   |
| 12 |  |   | ■ |   |   |
| 13 |  | ■ |   |   |   |
| 14 |  |   | ■ |   |   |
| 15 |  |   |   |   | ■ |
| 16 |  | ■ |   |   |   |

Risultato prova n. 33:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   |   |   |   |   |   |
| 2  |   |   |   |   |   |   |   |
| 3  |   |   | ■ | ■ |   |   |   |
| 4  |   |   |   |   | ■ | ■ |   |
| 5  |   |   |   | ■ | ■ |   |   |
| 6  | ■ | ■ |   |   |   |   |   |
| 7  | ■ | ■ |   |   |   |   |   |
| 8  | ■ | ■ |   |   |   |   |   |
| 9  |   |   |   |   | ■ | ■ |   |
| 10 |   |   |   | ■ | ■ |   |   |
| 11 |   |   | ■ | ■ |   |   |   |
| 12 |   |   |   | ■ | ■ |   |   |
| 13 |   |   |   |   |   | ■ | ■ |
| 14 | ■ | ■ |   |   |   |   |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 |   |   |   | ■ |   |   |   |

Risultato prova n. 34:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 35:

|    |  |   |   |   |   |   |  |
|----|--|---|---|---|---|---|--|
| 1  |  |   | ■ |   |   |   |  |
| 2  |  | ■ |   |   |   |   |  |
| 3  |  |   |   |   | ■ | ■ |  |
| 4  |  |   |   | ■ |   |   |  |
| 5  |  | ■ |   |   |   |   |  |
| 6  |  | ■ |   |   |   |   |  |
| 7  |  |   |   |   | ■ | ■ |  |
| 8  |  |   | ■ |   |   |   |  |
| 9  |  | ■ |   |   |   |   |  |
| 10 |  |   | ■ |   |   |   |  |
| 11 |  |   |   |   | ■ | ■ |  |
| 12 |  | ■ |   |   |   |   |  |
| 13 |  |   |   | ■ |   |   |  |
| 14 |  |   |   |   |   | ■ |  |
| 15 |  |   |   | ■ |   |   |  |
| 16 |  |   | ■ |   |   |   |  |

Risultato prova n. 36:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 37:

|    |  |   |   |   |   |   |  |
|----|--|---|---|---|---|---|--|
| 1  |  | ■ |   |   |   |   |  |
| 2  |  |   |   |   | ■ | ■ |  |
| 3  |  | ■ |   |   |   |   |  |
| 4  |  |   |   | ■ | ■ |   |  |
| 5  |  | ■ |   |   |   |   |  |
| 6  |  |   |   |   | ■ | ■ |  |
| 7  |  | ■ |   |   |   |   |  |
| 8  |  |   | ■ | ■ |   |   |  |
| 9  |  | ■ |   |   |   |   |  |
| 10 |  |   |   | ■ | ■ |   |  |
| 11 |  |   |   | ■ | ■ |   |  |
| 12 |  |   |   |   | ■ | ■ |  |
| 13 |  |   | ■ | ■ |   |   |  |
| 14 |  |   |   | ■ |   | ■ |  |
| 15 |  |   | ■ | ■ |   |   |  |
| 16 |  |   |   | ■ |   |   |  |

Risultato prova n. 38:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   | ■ | ■ |   |   |
| 3  |  |   | ■ | ■ |   |   |   |
| 4  |  | ■ | ■ |   |   |   |   |
| 5  |  |   |   |   | ■ | ■ |   |
| 6  |  |   |   |   | ■ | ■ |   |
| 7  |  | ■ | ■ |   |   |   |   |
| 8  |  | ■ | ■ |   |   |   |   |
| 9  |  |   | ■ | ■ |   |   |   |
| 10 |  |   |   | ■ | ■ |   |   |
| 11 |  | ■ | ■ |   |   |   |   |
| 12 |  |   |   | ■ | ■ |   |   |
| 13 |  |   | ■ | ■ |   |   |   |
| 14 |  |   |   | ■ | ■ |   |   |
| 15 |  |   |   |   |   | ■ | ■ |
| 16 |  |   |   |   | ■ | ■ |   |

Risultato prova n. 39:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 40:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 41:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 42:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 43:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  |   |   |   |   |   | 1 |
| 2  |  | 1 |   |   |   |   |   |
| 3  |  |   | 1 | 1 |   |   |   |
| 4  |  |   |   | 1 |   |   |   |
| 5  |  | 1 |   |   |   |   |   |
| 6  |  |   |   | 1 |   |   |   |
| 7  |  |   |   |   | 1 | 1 |   |
| 8  |  |   |   |   |   | 1 |   |
| 9  |  |   |   |   | 1 |   |   |
| 10 |  | 1 |   |   |   |   |   |
| 11 |  | 1 |   |   |   |   |   |
| 12 |  |   |   | 1 |   |   |   |
| 13 |  | 1 |   |   |   |   |   |
| 14 |  |   |   | 1 | 1 |   |   |
| 15 |  |   |   |   | 1 |   |   |
| 16 |  |   |   |   |   | 1 |   |

Risultato prova n. 44:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 45:

|    |  |   |   |   |   |   |  |
|----|--|---|---|---|---|---|--|
| 1  |  | ■ |   |   |   |   |  |
| 2  |  |   | ■ |   |   |   |  |
| 3  |  | ■ |   |   |   |   |  |
| 4  |  |   |   | ■ | ■ |   |  |
| 5  |  |   |   | ■ | ■ |   |  |
| 6  |  |   |   | ■ | ■ |   |  |
| 7  |  | ■ |   |   |   |   |  |
| 8  |  | ■ |   |   |   |   |  |
| 9  |  |   |   |   |   | ■ |  |
| 10 |  |   | ■ |   |   |   |  |
| 11 |  |   |   |   | ■ | ■ |  |
| 12 |  |   |   |   | ■ | ■ |  |
| 13 |  |   | ■ |   |   |   |  |
| 14 |  | ■ |   |   |   |   |  |
| 15 |  |   |   |   | ■ | ■ |  |
| 16 |  |   |   | ■ |   |   |  |

Risultato prova n. 46:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  |   |   | ■ | ■ |   |   |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  |   |   |   |   | ■ | ■ |   |
| 6  |   |   |   |   | ■ | ■ |   |
| 7  |   | ■ | ■ |   |   |   |   |
| 8  |   |   |   | ■ | ■ |   |   |
| 9  |   |   | ■ | ■ |   |   |   |
| 10 |   |   |   | ■ | ■ |   |   |
| 11 |   |   |   |   |   | ■ | ■ |
| 12 |   | ■ | ■ |   |   |   |   |
| 13 |   |   |   | ■ | ■ |   |   |
| 14 |   | ■ | ■ |   |   |   |   |
| 15 |   |   |   | ■ | ■ |   |   |
| 16 |   |   | ■ | ■ |   |   |   |

Risultato prova n. 47:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   | ■ | ■ |   |   |   |
| 3  |  |   | ■ | ■ |   |   |   |
| 4  |  |   |   |   | ■ | ■ |   |
| 5  |  | ■ | ■ |   |   |   |   |
| 6  |  |   | ■ | ■ |   |   |   |
| 7  |  |   |   |   | ■ | ■ |   |
| 8  |  |   |   | ■ | ■ |   |   |
| 9  |  |   |   |   |   | ■ | ■ |
| 10 |  |   |   | ■ | ■ |   |   |
| 11 |  |   |   |   | ■ | ■ |   |
| 12 |  | ■ | ■ |   |   |   |   |
| 13 |  | ■ | ■ |   |   |   |   |
| 14 |  |   |   | ■ | ■ |   |   |
| 15 |  | ■ | ■ |   |   |   |   |
| 16 |  |   |   | ■ | ■ |   |   |

Risultato prova n. 48:

|    |   |   |   |   |  |
|----|---|---|---|---|--|
| 1  |   |   | ■ |   |  |
| 2  |   | ■ |   |   |  |
| 3  |   |   |   | ■ |  |
| 4  |   |   |   | ■ |  |
| 5  |   |   | ■ |   |  |
| 6  | ■ |   |   |   |  |
| 7  |   |   |   | ■ |  |
| 8  | ■ |   |   |   |  |
| 9  | ■ |   |   |   |  |
| 10 | ■ |   |   |   |  |
| 11 |   |   |   | ■ |  |
| 12 |   |   | ■ |   |  |
| 13 |   | ■ |   |   |  |
| 14 | ■ |   |   |   |  |
| 15 |   | ■ |   |   |  |
| 16 |   |   | ■ |   |  |

Risultato prova n. 49:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 50:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   | ■ | ■ |   |   |
| 3  |  |   | ■ | ■ |   |   |   |
| 4  |  |   |   |   |   | ■ | ■ |
| 5  |  | ■ | ■ |   |   |   |   |
| 6  |  |   | ■ | ■ |   |   |   |
| 7  |  |   |   | ■ | ■ |   |   |
| 8  |  | ■ | ■ |   |   |   |   |
| 9  |  |   |   | ■ | ■ |   |   |
| 10 |  | ■ | ■ |   |   |   |   |
| 11 |  |   |   | ■ | ■ |   |   |
| 12 |  |   |   |   | ■ | ■ |   |
| 13 |  |   | ■ | ■ |   |   |   |
| 14 |  |   |   |   | ■ | ■ |   |
| 15 |  |   |   |   |   | ■ | ■ |
| 16 |  | ■ | ■ |   |   |   |   |

Risultato prova n. 51:

|    |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|--|
| 1  |   |   | ■ |   |   |   |  |
| 2  |   |   |   | ■ |   |   |  |
| 3  | ■ |   |   |   |   |   |  |
| 4  | ■ |   |   |   |   |   |  |
| 5  | ■ |   |   |   |   |   |  |
| 6  | ■ |   |   |   |   |   |  |
| 7  |   |   |   |   | ■ |   |  |
| 8  |   |   |   | ■ |   |   |  |
| 9  |   |   | ■ |   |   |   |  |
| 10 |   |   |   |   | ■ |   |  |
| 11 | ■ |   |   |   |   |   |  |
| 12 |   | ■ |   |   |   |   |  |
| 13 |   |   |   |   |   | ■ |  |
| 14 |   |   | ■ |   |   |   |  |
| 15 |   |   |   |   | ■ |   |  |
| 16 |   |   | ■ |   |   |   |  |

Risultato prova n. 52:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   |   | ■ | ■ |   |
| 3  |  |   | ■ | ■ |   |   |   |
| 4  |  |   |   |   | ■ | ■ |   |
| 5  |  |   |   | ■ | ■ |   |   |
| 6  |  |   |   |   | ■ | ■ |   |
| 7  |  | ■ | ■ |   |   |   |   |
| 8  |  | ■ |   |   |   |   |   |
| 9  |  |   |   |   | ■ | ■ |   |
| 10 |  |   |   |   | ■ | ■ |   |
| 11 |  | ■ |   |   |   |   |   |
| 12 |  |   | ■ | ■ |   |   |   |
| 13 |  |   |   |   |   | ■ | ■ |
| 14 |  | ■ | ■ |   |   |   |   |
| 15 |  |   |   |   |   | ■ | ■ |
| 16 |  |   |   |   |   |   | ■ |

Risultato prova n. 53:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 54:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 55:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 56:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  | ■ | ■ |   |   |   |   |   |
| 3  |   |   |   |   | ■ | ■ |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  | ■ | ■ |   |   |   |   |   |
| 6  |   |   |   |   | ■ | ■ |   |
| 7  |   |   |   | ■ | ■ |   |   |
| 8  |   |   | ■ | ■ |   |   |   |
| 9  |   |   |   | ■ | ■ |   |   |
| 10 | ■ | ■ |   |   |   |   |   |
| 11 |   |   |   |   |   | ■ | ■ |
| 12 | ■ | ■ |   |   |   |   |   |
| 13 |   |   |   | ■ | ■ |   |   |
| 14 |   |   | ■ | ■ |   |   |   |
| 15 |   |   |   |   | ■ | ■ |   |
| 16 |   |   |   | ■ | ■ |   |   |

Risultato prova n. 57:

|    |  |  |  |  |  |  |    |
|----|--|--|--|--|--|--|----|
| 1  |  |  |  |  |  |  | 1  |
| 2  |  |  |  |  |  |  | 2  |
| 3  |  |  |  |  |  |  | 3  |
| 4  |  |  |  |  |  |  | 4  |
| 5  |  |  |  |  |  |  | 5  |
| 6  |  |  |  |  |  |  | 6  |
| 7  |  |  |  |  |  |  | 7  |
| 8  |  |  |  |  |  |  | 8  |
| 9  |  |  |  |  |  |  | 9  |
| 10 |  |  |  |  |  |  | 10 |
| 11 |  |  |  |  |  |  | 11 |
| 12 |  |  |  |  |  |  | 12 |
| 13 |  |  |  |  |  |  | 13 |
| 14 |  |  |  |  |  |  | 14 |
| 15 |  |  |  |  |  |  | 15 |
| 16 |  |  |  |  |  |  | 16 |

Risultato prova n. 58:

|    |   |  |   |   |   |   |  |
|----|---|--|---|---|---|---|--|
| 1  |   |  | ■ |   |   |   |  |
| 2  | ■ |  |   |   |   |   |  |
| 3  |   |  |   |   |   | ■ |  |
| 4  |   |  | ■ |   |   |   |  |
| 5  |   |  |   |   | ■ |   |  |
| 6  |   |  | ■ |   |   |   |  |
| 7  |   |  |   | ■ |   |   |  |
| 8  |   |  |   | ■ |   |   |  |
| 9  | ■ |  |   |   |   |   |  |
| 10 | ■ |  |   |   |   |   |  |
| 11 | ■ |  |   |   |   |   |  |
| 12 |   |  |   | ■ |   |   |  |
| 13 |   |  |   |   | ■ |   |  |
| 14 | ■ |  |   |   |   |   |  |
| 15 |   |  |   |   | ■ |   |  |
| 16 |   |  |   | ■ |   |   |  |

Risultato prova n. 59:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 60:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 61:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 62:

|    |   |   |   |   |   |   |  |  |
|----|---|---|---|---|---|---|--|--|
| 1  |   | ■ |   |   |   |   |  |  |
| 2  |   |   | ■ |   |   |   |  |  |
| 3  |   |   |   | ■ |   |   |  |  |
| 4  |   | ■ |   |   |   |   |  |  |
| 5  |   |   |   |   | ■ |   |  |  |
| 6  |   |   |   | ■ |   |   |  |  |
| 7  |   |   |   |   | ■ |   |  |  |
| 8  |   |   |   |   |   | ■ |  |  |
| 9  |   |   |   | ■ |   |   |  |  |
| 10 |   | ■ |   |   |   |   |  |  |
| 11 |   |   | ■ |   |   |   |  |  |
| 12 |   |   |   |   | ■ |   |  |  |
| 13 |   | ■ |   |   |   |   |  |  |
| 14 |   |   |   | ■ |   |   |  |  |
| 15 |   |   | ■ |   |   |   |  |  |
| 16 | ■ |   |   |   |   |   |  |  |

Risultato prova n. 63:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   | ■ | ■ |   |   |
| 3  |   | ■ |   |   |   |   |   |
| 4  |   |   | ■ |   |   |   |   |
| 5  |   |   |   |   | ■ | ■ |   |
| 6  |   |   |   |   |   | ■ | ■ |
| 7  |   |   |   |   | ■ | ■ |   |
| 8  |   |   |   | ■ | ■ |   |   |
| 9  |   |   |   |   | ■ | ■ |   |
| 10 |   |   | ■ | ■ |   |   |   |
| 11 |   |   | ■ | ■ |   |   |   |
| 12 |   |   |   |   | ■ | ■ |   |
| 13 |   | ■ | ■ |   |   |   |   |
| 14 |   | ■ | ■ |   |   |   |   |
| 15 |   |   |   | ■ | ■ |   |   |
| 16 | ■ | ■ |   |   |   |   |   |

Risultato prova n. 64:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 65:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 66:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 67:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 68:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 69:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  | ■ |   |   |   |   |   |   |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  |   |   |   |   |   | ■ | ■ |
| 5  |   |   |   |   |   | ■ | ■ |
| 6  | ■ | ■ |   |   |   |   |   |
| 7  | ■ | ■ |   |   |   |   |   |
| 8  |   |   |   | ■ | ■ |   |   |
| 9  | ■ | ■ |   |   |   |   |   |
| 10 | ■ | ■ |   |   |   |   |   |
| 11 |   |   | ■ | ■ |   |   |   |
| 12 |   |   |   |   | ■ | ■ |   |
| 13 |   |   |   |   |   |   | ■ |
| 14 |   |   | ■ | ■ |   |   |   |
| 15 |   |   |   | ■ | ■ |   |   |
| 16 |   |   |   | ■ | ■ |   |   |

Risultato prova n. 70:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 71:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 72:

|    |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|--|
| 1  |   | ■ |   |   |   |   |  |
| 2  |   |   |   | ■ |   |   |  |
| 3  |   |   |   |   |   | ■ |  |
| 4  | ■ |   |   |   |   |   |  |
| 5  |   |   | ■ |   |   |   |  |
| 6  | ■ |   |   |   |   |   |  |
| 7  |   |   | ■ |   |   |   |  |
| 8  |   |   |   | ■ |   |   |  |
| 9  |   |   |   | ■ |   |   |  |
| 10 |   |   |   | ■ |   |   |  |
| 11 |   |   |   |   | ■ |   |  |
| 12 | ■ |   |   |   |   |   |  |
| 13 |   |   |   |   | ■ |   |  |
| 14 |   |   | ■ |   |   |   |  |
| 15 | ■ |   |   |   |   |   |  |
| 16 |   |   |   |   | ■ |   |  |

Risultato prova n. 73:

|    |   |  |   |   |   |   |  |
|----|---|--|---|---|---|---|--|
| 1  |   |  | ■ |   |   |   |  |
| 2  |   |  |   | ■ |   |   |  |
| 3  | ■ |  |   |   |   |   |  |
| 4  |   |  |   |   | ■ |   |  |
| 5  |   |  |   | ■ |   |   |  |
| 6  |   |  |   |   | ■ |   |  |
| 7  |   |  |   |   |   | ■ |  |
| 8  |   |  |   |   | ■ |   |  |
| 9  | ■ |  |   |   |   |   |  |
| 10 |   |  | ■ |   |   |   |  |
| 11 | ■ |  |   |   |   |   |  |
| 12 | ■ |  |   |   |   |   |  |
| 13 |   |  |   | ■ |   |   |  |
| 14 | ■ |  |   |   |   |   |  |
| 15 |   |  | ■ |   |   |   |  |
| 16 |   |  |   | ■ |   |   |  |

Risultato prova n. 74:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 75:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 76:

|    |  |   |   |   |   |   |    |
|----|--|---|---|---|---|---|----|
| 1  |  |   |   |   |   |   | 1  |
| 2  |  | 1 |   |   |   |   | 2  |
| 3  |  |   |   |   |   | 1 | 3  |
| 4  |  |   | 1 |   |   |   | 4  |
| 5  |  |   |   | 1 |   |   | 5  |
| 6  |  |   |   |   | 1 |   | 6  |
| 7  |  |   | 1 |   |   |   | 7  |
| 8  |  |   |   | 1 |   |   | 8  |
| 9  |  |   |   |   | 1 |   | 9  |
| 10 |  | 1 |   |   |   |   | 10 |
| 11 |  | 1 |   |   |   |   | 11 |
| 12 |  | 1 |   |   |   |   | 12 |
| 13 |  |   |   | 1 |   |   | 13 |
| 14 |  |   |   |   | 1 |   | 14 |
| 15 |  | 1 |   |   |   |   | 15 |
| 16 |  |   | 1 |   |   |   | 16 |

Risultato prova n. 77:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   |   |   |   | ■ |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  | ■ | ■ |   |   |   |   |   |
| 6  |   |   |   |   | ■ | ■ |   |
| 7  | ■ | ■ |   |   |   |   |   |
| 8  |   |   | ■ | ■ |   |   |   |
| 9  |   |   | ■ | ■ |   |   |   |
| 10 | ■ | ■ |   |   |   |   |   |
| 11 |   |   |   | ■ | ■ |   |   |
| 12 |   |   |   | ■ | ■ |   |   |
| 13 |   |   | ■ | ■ |   |   |   |
| 14 |   |   |   | ■ | ■ |   |   |
| 15 |   |   |   |   | ■ | ■ |   |
| 16 |   |   |   |   | ■ | ■ |   |

Risultato prova n. 78:

|    |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|--|
| 1  |   |   | ■ |   |   |   |  |
| 2  |   |   |   | ■ |   |   |  |
| 3  |   |   |   |   |   | ■ |  |
| 4  |   |   | ■ |   |   |   |  |
| 5  |   |   |   |   | ■ |   |  |
| 6  |   |   | ■ |   |   |   |  |
| 7  | ■ | ■ |   |   |   |   |  |
| 8  | ■ | ■ |   |   |   |   |  |
| 9  | ■ | ■ |   |   |   |   |  |
| 10 | ■ | ■ |   |   |   |   |  |
| 11 |   |   |   |   | ■ |   |  |
| 12 |   |   |   | ■ | ■ |   |  |
| 13 |   |   |   |   | ■ | ■ |  |
| 14 | ■ | ■ |   |   |   |   |  |
| 15 |   |   |   | ■ | ■ |   |  |
| 16 |   |   |   | ■ | ■ |   |  |

Risultato prova n. 79:

|    |   |   |   |   |   |   |  |   |
|----|---|---|---|---|---|---|--|---|
| 1  |   |   | ■ |   |   |   |  |   |
| 2  |   |   |   | ■ |   |   |  |   |
| 3  |   |   |   |   | ■ |   |  |   |
| 4  |   |   |   |   |   |   |  | ■ |
| 5  |   |   |   | ■ |   |   |  |   |
| 6  |   |   |   |   | ■ |   |  |   |
| 7  |   |   |   |   | ■ |   |  |   |
| 8  |   |   |   |   |   | ■ |  |   |
| 9  |   |   |   |   |   | ■ |  |   |
| 10 |   | ■ | ■ |   |   |   |  |   |
| 11 |   | ■ | ■ |   |   |   |  |   |
| 12 |   |   |   |   |   | ■ |  |   |
| 13 |   | ■ | ■ |   |   |   |  |   |
| 14 |   | ■ | ■ |   |   |   |  |   |
| 15 |   |   |   |   | ■ |   |  |   |
| 16 | ■ | ■ |   |   |   |   |  |   |

Risultato prova n. 80:



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 1**

1. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception { }
```

```

try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

5. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
E. Nessuna delle precedenti.
- 

6. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

7. Qual'è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

8. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}

```

```

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {

```

```

Boolean b1 = new Boolean(true);
if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;
- 

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
    }
}

```

```

c1 = null;
c3 = c1;
/* QUI */
return;
}
}
A. 1
B. 2
C. 3
D. 4
E. 6

```

---

15. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
        }
    }
}

```

```
        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 2**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    }
}
```

```
/* QUI */
return;
}
A. 1
B. 2
C. 3
D. 4
E. 6
```

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
- B. 1

- C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

6. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}
```

```
class House implements Building{  
    String type = "Residential";
```

```

int age;
public int getAge() {
    return age;
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 10. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {

```

```

protected void f(Z z) {
    System.out.print("Z");
}
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 11. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

#### 12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

14. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
- 

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {

```

```
        throw( new MyExc1() ) ;
    }
}

static void m()
throws Exception {
    try {
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}
```

- 
- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 3**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
```

```
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

4. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
}
```

```

public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

6. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

9. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
                m();
            }
```

```
} catch( Exception s ) {
    System.out.print(2);
    throw( new MyExc1() );
}
catch( MyExc2 z ) {
    System.out.print(3);
}
finally {
    throw( new MyExc1() );
}
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

12. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

**14. Dato un file contenente il seguente codice:**

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

**15. Qual è l'output di questo programma?**

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

16. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
}
```

```
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 4

1. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc3 b ) {  
        }  
        catch( MyExc2 j ) {  
        }  
        catch( Exception v ) {  
            System.out.print(2);  
        }  
        finally {  
            throw( new MyExc2() );  
        }  
    }  
    static void m()  
        throws Exception {  
        try {  
        }  
        catch( Exception g ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;
```

```

Coffee() {}
Coffee(int ml) {
    this.ml = ml;
}
public static void main(String args[]) {
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata `abstract` o `final`
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

6. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125

- B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new O.I()`.
- E. Nessuna delle precedenti.

---

9. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0

D. null

E. Nessuna delle precedenti

---

11. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 z ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc1() );  
            }  
        }  
        static void m()  
        throws Exception {  
            try {  
            }  
            catch( Exception h ) {  
            }  
            finally {  
                System.out.print(4);  
            }  
        }  
    }
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1

- C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

14. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

15. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

16. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
          Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(3);  
        }  
    }  
}
```

```
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- 
- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 5**

1. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

2. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
} }
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

3. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}

class Mammal extends Animal{
    Mammal(int i) {
```

```

        super(6);
        System.out.print(5);
    }
}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

---

9. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
- B. 12222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

11. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
E. Nessuna delle precedenti.

---

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6
- 

13. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- 
- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 6**

1. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- ```
    }
}

A. 12Exception in thread "main" MyExc3
B. 1222222... (ciclo infinito)
C. 12
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
```

```

}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

6. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `x = (Error []) y;`
  - B. `x = y;`
  - C. `y = m;`
  - D. `x = m;`
  - E. `m = (Object []) x;`
- 

9. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
        }
    }
}
```

```
        throw( new MyExc1() );
    }
    catch( MyExc2 z ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
    try {
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

14. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

15. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{

```

```
Mammal() {  
    super(6);  
    System.out.print(5);  
}  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 7**

1. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
```

```
public static void main(String[] args) {
    X x = new Z();
    Y y = (Y)x;
    Z z = (Z)y;
    z.f(x); z.f(y); z.f(z);
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- B. 101
  - C. 100
  - D. 011
  - E. 010
- 

4. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

5. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
```

```
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

9. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

A. 1
B. 2
C. 3
D. 4
E. 6
```

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

12. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

15. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

16. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}
```

```
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- ```
    }  
  
A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
```
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 8**

1. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

2. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
-

---

4. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 z ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc1() );  
            }  
        }  
        static void m()  
            throws Exception {  
                try {  
                }  
                catch( Exception h ) {  
                }  
                finally {  
                    System.out.print(4);  
                }  
            }  
    }
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
}  
public static void main(String args[]) {  
    House h = new House();  
}
```

```

        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

11. Qual è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 9**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}
```

```
class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
 B. XYZ  
 C. ZZZ  
 D. Errore a tempo di compilazione.  
 E. Nessuna delle precedenti.
- 

5. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
    }
}

```

- z.f(x); z.f(y); z.f(z);  
 }  
 A. XXX  
 B. XYZ  
 C. ZZZ  
 D. Errore a tempo di compilazione.  
 E. Nessuna delle precedenti.
- 

6. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. 6
- 

7. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
 B. x = y;  
 C. y = m;  
 D. x = m;  
 E. m = (Object []) x;
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        } catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        } catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 9. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 10. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;  
B. g = (Boolean) s;  
C. b = (Integer) g;  
D. g = (Boolean) b;  
E. Nessuno dei precedenti
- 

#### 11. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
E. Nessuna delle precedenti.
- 

#### 12. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}

```

```

}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

14. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {

```

```

            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {

```

```
try {  
}  
catch( Exception g ) {  
}  
finally {  
    System.out.print(3);  
}  
}  
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 10**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
```

```
} }
```

```
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
```

```

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;

```
E. m = (Object []) x;
```

---

6. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {

```

```

        System.out.print(3);
    }
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

8. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
E. Nessuna delle precedenti.
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
B. Una classe può essere dichiarata abstract o final  
C. Due metodi non possono differire per il tipo di ritorno  
D. Un array non possiede dei membri  
E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] args) {
        try {
            p();
        }
    }
}

```

```

        catch( MyExc2 r ) {
    }
}
static void p() {
    try {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}

```

---

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

**13. Qual è l'output di questo codice?**

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**14. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?**

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

**15. Qual è l'output di questo codice?**

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

16. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 11**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

```
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

---

4. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building {  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0

D. null

E. Nessuna delle precedenti

---

6. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal {  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]) {  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3

- D. 4  
E. 6
- 

8. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
B. Una classe può essere dichiarata **abstract** o **final**  
C. Due metodi non possono differire per il tipo di ritorno  
D. Un array non possiede dei membri  
E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();
```

```
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( Exception s ) {  
            System.out.print(2);  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 z ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc1() );  
        }  
    }  
    static void m()  
        throws Exception {  
        try {  
        }  
        catch( Exception h ) {  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)

E. Nessuna delle precedenti

---

12. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

13. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

14. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

---

15. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
    finally {

```

```
        throw( new MyExc2() );
    }
}

static void m()
throws Exception {
    try {
    }
    catch( Exception g ) {
    }
    finally {
        System.out.print(3);
    }
}
```

- }
- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 12

1. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
    }}
```

```
}
```

```
static void p() {
```

```
try {
    System.out.print(1);
    throw( new MyExc3() );
}
```

```
catch( MyExc3 y ) {
```

```
System.out.print(2);
throw( new MyExc3() );
}
```

```
}
```

```
}
```

- A. 12Exception in thread "main" MyExc3

- B. 1222222... (ciclo infinito)

- C. 12

- D. Errore a tempo di compilazione

- E. Nessuna delle precedenti

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1

- B. 2

- C. 3

- D. 4

- E. 6

---

5. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

6. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( Exception s ) {  
            System.out.print(2);  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 z ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc1() );  
        }  
    }  
    static void m()  
        throws Exception {  
        try {
```

```

    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {

```

```

        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

11. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
-

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
```

```
public static void main(String args[]) {
    System.out.print(1);
    Mammal m = new Mammal();
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

15. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-

16. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 13**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
```

```
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
```

```

        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

5. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

6. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;
}

```

```

Coffee() {}
Coffee(int ml){
    this.ml = ml;
}
public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}
A. 1
B. 2
C. 3
D. 4
E. 6

```

---

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

9. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

10. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

11. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;

E. m = (Object []) x;

---

12. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 z ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc1() );  
            }  
        }  
        static void m()  
            throws Exception {
```

```
        try {  
        }  
        catch( Exception h ) {  
        }  
        finally {  
            System.out.print(4);  
        }  
    }
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

15. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 14**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
```

```
public static void main(String args[]) {
    System.out.print(1);
    Mammal m = new Mammal();
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

3. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata `abstract` o `final`
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

4. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

5. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

6. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

---

E. Nessuna delle precedenti.

---

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
    }
}

```

```

        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {

```

```

        System.out.print(1);
        m();
    }
    catch( Exception s ) {
        System.out.print(2);
        throw( new MyExc1() );
    }
    catch( MyExc2 z ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
    try {
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

### 13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
            try {

```

```

        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

### 14. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;  
B. g = (Boolean) s;  
C. b = (Integer) g;  
D. g = (Boolean) b;  
E. Nessuno dei precedenti
- 

### 15. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;
- 

### 16. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 15**

1. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

3. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

4. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

5. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
}
public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
```

```
        return;
    }
}

A. 1
B. 2
C. 3
D. 4
E. 6
```

---

7. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

8. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```

    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `x = (Error []) y;`
  - B. `x = y;`
  - C. `y = m;`
  - D. `x = m;`
  - E. `m = (Object []) x;`
- 

13. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 16**

1. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
                m();
            }
            catch( Exception s ) {
                System.out.print(2);
                throw( new MyExc1() );
            }
        }
}
```

```
catch( MyExc2 z ) {
    System.out.print(3);
}
finally {
    throw( new MyExc1() );
}
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
```

```

public static void main(String[] args) {
    X x = new Z();
    Y y = (Y)x;
    Z z = (Z)y;
    z.f(x); z.f(y); z.f(z);
}
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

5. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

6. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

**9. Qual è l'output di questo codice?**

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**10. Qual è l'output di questo codice?**

```

public interface Building {
    int age;
    public int getAge();
}

```

```

}
class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**11. Qual è l'output di questo codice?**

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**12. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
    }
}

```

```

        catch( MyExc2 j ) {
    }
    catch( Exception v ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
}

A. 13
B. 1
C. 13Exception in thread "main" MyExc2
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;

---

14. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

A. 125
B. 135
C. 1235
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

15. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 
16. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 17

1. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

2. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

3. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

4. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno

- D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

## 9. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {

```

```

            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

## 10. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

## 11. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0

- D. null
  - E. Nessuna delle precedenti
- 

12. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}  
  
A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
```

---

13. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {
```

```
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

15. Qual'è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- 
- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 18**

1. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

- E. Nessuna delle precedenti.
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
```

```

    }
}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

6. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;

```

```

g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

7. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. 0

D. null

E. Nessuna delle precedenti

---

#### 11. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

#### 12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1

C. 1

D. 14222222... (ciclo infinito)

E. Nessuna delle precedenti

- 
13. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

A. 1

B. 2

C. 3

D. 4

E. 6

- 
14. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

A. 125

B. 135

C. 1235

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

- 
15. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

- 
16. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        }  
    }  
}
```

```
} else {
    System.out.print(0);
}
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 19**

1. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
-

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml) {  
        this.ml = ml;  
    }  
    public static void main(String args[]) {  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

- 
5. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

- 
6. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

- 
7. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata **abstract** o **final**
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

- 
8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc3 b ) {  
        }  
        catch( MyExc2 j ) {  
        }  
        catch( Exception v ) {  
            System.out.print(2);  
        }  
        finally {  
            throw( new MyExc2() );  
        }  
    }  
}
```

```

    }
}

static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1

- C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

#### 11. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
E. Nessuna delle precedenti.
- 

#### 12. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
 B. x = y;  
 C. y = m;  
 D. x = m;  
 E. m = (Object []) x;
- 

14. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {

```

```

        House h = new House();
        System.out.println(h.getAge());
    }
}

A. Errore a tempo di compilazione  

B. Errore a tempo di esecuzione  

C. 0  

D. null  

E. Nessuna delle precedenti

```

---

15. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX  

B. XYZ  

C. ZZZ  

D. Errore a tempo di compilazione.  

E. Nessuna delle precedenti.

```

---

16. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {

```

```
}
```

```
static void p() {
```

```
    try {
```

```
        System.out.print(1);
```

```
        throw( new MyExc3() );
```

```
    }
```

```
    catch( MyExc3 y ) {
```

```
        System.out.print(2);
```

```
        throw( new MyExc3() );
```

```
}
```

```
}
```

```
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 20**

1. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
```

```
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
```

```

        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 5. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

#### 6. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

```

```

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

#### 7. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010

---

8. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
}
public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}
```

```
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

13. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {  
            }  
            catch( MyExc2 j ) {  
            }  
            catch( Exception v ) {  
                System.out.print(2);  
            }  
            finally {  
                throw( new MyExc2() );  
            }  
        }  
        static void m()  
            throws Exception {  
            try {  
            }  
            catch( Exception g ) {  
            }  
            finally {  
                System.out.print(3);  
            }  
        }  
}
```

- A. 13
- B. 1

- C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 21**

1. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0

D. null

E. Nessuna delle precedenti

---

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
```

```

        System.out.print(2);
        throw( new MyExc3() );
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse

- B. Una classe può essere dichiarata **abstract** o **final**  
C. Due metodi non possono differire per il tipo di ritorno  
D. Un array non possiede dei membri  
E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

#### 7. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

9. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
        }
```

```
        m();
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc2 j ) {
    }
    catch( Exception v ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

14. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

15. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}
```

```
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 22**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}
```

```
} }
```

```
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}
```

```
class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
}
```

```
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

6. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {
```

```
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

9. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

14. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

```

```

    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {                                     /* QUI */
    String brand ="Nespresso";
    static Integer ml = 20;
}
Coffee() {}
Coffee(int ml){
    this.ml = ml;
}
public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
```

---

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 23**

1. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}

class Mammal extends Animal{
```

```
Mammal() {
    super(6);
    System.out.print(5);
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

3. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX

- B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

4. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

5. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
}
```

```

    }
    public static void main(String args[]) {
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13

- B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
        }
    }
}

```

```
        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 24**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
-

4. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

5. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

6. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null

E. Nessuna delle precedenti

7. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1

- D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

9. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee(){}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
    }  
}
```

```
        return;  
    }  
}  
  
A. 1  
B. 2  
C. 3  
D. 4  
E. 6
```

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

15. Qual è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

16. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
```

```
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

A. 125

- B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 25**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        }
    }
}
```

```
} else {
    System.out.print(0);
}
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125

- B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

5. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

6. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

7. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
            }
```

```

        m();
    }
    catch( Exception s ) {
        System.out.print(2);
        throw( new MyExc1() );
    }
    catch( MyExc2 z ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

11. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {

```

```

}
catch( MyExc2 j ) {
}
catch( Exception v ) {
    System.out.print(2);
}
finally {
    throw( new MyExc2() );
}
}

static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
}
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }

    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

14. Qual è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

15. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0

- D. null
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
-



---

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 26**

1. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
            }
```

```
    m();
}
catch( Exception s ) {
    System.out.print(2);
    throw( new MyExc1() );
}
catch( MyExc2 z ) {
    System.out.print(3);
}
finally {
    throw( new MyExc1() );
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
                m();
            }
            catch( MyExc3 b ) {
            }
            catch( MyExc2 j ) {
```

```

}
catch( Exception v ) {
    System.out.print(2);
}
finally {
    throw( new MyExc2() );
}
}

static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

5. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
E. Nessuna delle precedenti.
- 

6. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
}

```

```

void p() {
    Boolean b3 = new Boolean(true);
    s7 = s6;
    q(b3, s2, s7, s6);
}
void q(Boolean b2, String s3, String s4,
Object s5) {
    Boolean b1 = new Boolean(true);
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

7. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
-

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml) {  
        this.ml = ml;  
    }  
    public static void main(String args[]) {  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

9. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

10. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {
```

```
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

11. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

14. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
}
```

```
Animal(int i) {
    System.out.print(3);
}
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 27**

1. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

2. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

3. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

5. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

7. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

8. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135

C. 1235

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

9. Dire quale delle seguenti affermazioni è vera:

A. Un array ha più di una superclasse

B. Una classe può essere dichiarata **abstract** o **final**

C. Due metodi non possono differire per il tipo di ritorno

D. Un array non possiede dei membri

E. Ogni valore in memoria non è associato ad un tipo di dato particolare

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. Errore a tempo di compilazione

B. 14Exception in thread "main" MyExc1

C. 1

D. 14222222... (ciclo infinito)

E. Nessuna delle precedenti

---

11. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.

D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con **new O.I()**.

E. Nessuna delle precedenti.

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

A. 13

- B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;
```

```
public int getAge() {  
    return age;  
}  
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}  
}
```

- 
- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 28**

1. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

3. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `x = (Error []) y;`
- B. `x = y;`
- C. `y = m;`
- D. `x = m;`
- E. `m = (Object []) x;`

---

6. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 z ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc1() );  
            }  
        }  
        static void m()  
            throws Exception {  
            try {  
            }  
            catch( Exception h ) {  
            }  
            finally {  
                System.out.print(4);  
            }  
        }  
    }
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;
```

```

public int getAge() {
    return age;
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
}

```

```

public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

---

12. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

13. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

14. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

15. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
    finally {
        throw( new MyExc2() );
    }
}

```

```
        }
    }
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
```

- ```
        }
    }
A. 13
B. 1
C. 13Exception in thread "main" MyExc2
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 29**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
```

```
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

4. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;

- C. `y = m;`
  - D. `x = m;`
  - E. `m = (Object []) x;`
- 

5. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

7. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
```

```
}
```

```
}
```

```
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
```

```
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
```

```
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
```

```
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 14. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 15. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

#### 16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 30**

1. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
```

```
X x = new Z();
Y y = new Z();
Z z = new Z();
z.f(x); z.f(y); z.f(z);
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
```

```

Mammal(int i) {
    super(6);
    System.out.print(5);
}
}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 5. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}

class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {

```

```

            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
-

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        }
    }
}
```

```
    } else {
        System.out.print(0);
    }
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

10. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

11. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

### 13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;
- 

### 14. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

### 15. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

### 16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
}

```

```
}

public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}
```

```
}

A. 1
B. 2
C. 3
D. 4
E. 6
```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 31**

1. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

3. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
```

```
private void f(Y y) {
    System.out.print("Y");
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

5. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

6. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {  
            }
```

```
                catch( MyExc2 j ) {  
                }  
                catch( Exception v ) {  
                    System.out.print(2);  
                }  
                finally {  
                    throw( new MyExc2() );  
                }  
            }  
        static void m()  
            throws Exception {  
                try {  
                }  
                catch( Exception g ) {  
                }  
                finally {  
                    System.out.print(3);  
                }  
            }  
        }
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        } catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        } catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
 B. 1222222... (ciclo infinito)  
 C. 12  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
    throws Exception {
    try {
    }
    catch( Exception h ) {
    }
}

```

```

        finally {
            System.out.print(4);
        }
    }

```

- A. Errore a tempo di compilazione  
 B. 14Exception in thread "main" MyExc1  
 C. 1  
 D. 14222222... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

#### 11. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
 B. 101  
 C. 100  
 D. 011  
 E. 010
- 

#### 12. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

14. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

```

```
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 32**

1. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
    finally {
        throw( new MyExc2() );
    }
    static void m()
```

```

throws Exception {
    try {
    }
    catch( Exception g ) {
    }
    finally {
        System.out.print(3);
    }
}

```

- A. 13  
 B. 1  
 C. 13Exception in thread "main" MyExc2  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
 B. Una classe può essere dichiarata **abstract** o **final**  
 C. Due metodi non possono differire per il tipo di ritorno  
 D. Un array non possiede dei membri  
 E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

6. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
 B. XYZ  
 C. ZZZ  
 D. Errore a tempo di compilazione.  
 E. Nessuna delle precedenti.
- 

7. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

9. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
          Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

11. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

12. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- ```
}
```
- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

16. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
```

```
}

}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 33**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
```

```
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Dato un file contenente il seguente codice:

```
package pk;
public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

---

4. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ

C. ZZZ

- D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

7. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
    }  
}
```

```

        c1 = null;
        c3 = c1;
    /* QUI */
    return;
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6

#### 9. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

#### 10. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;

E. m = (Object []) x;

#### 11. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;  
B. g = (Boolean) s;  
C. b = (Integer) g;  
D. g = (Boolean) b;  
E. Nessuno dei precedenti

#### 12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011

## 13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

## 14. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

```

```

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

## 15. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

## 16. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}

class Mammal extends Animal{
    Mammal(int i) {

```

```
super(6);
System.out.print(5);
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

---

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti



prof. Piero A. Bonatti

26 giugno 2019

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 34

1. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

3. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
    }  
}
```

```
    Mammal m = new Mammal();  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

5. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 12222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {}  
            catch( MyExc2 j ) {}  
            catch( Exception v ) {
```

```
                System.out.print(2);  
            }  
            finally {  
                throw( new MyExc2() );  
            }  
        }  
        static void m()  
            throws Exception {  
                try {  
                }  
                catch( Exception g ) {  
                }  
                finally {  
                    System.out.print(3);  
                }  
            }  
        }  
    }
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal {  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

13. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 101
- C. 100
- D. 011

E. 010

14. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

16. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 35**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
```

```
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2

C. 3

D. 4

E. 6

---

4. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

5. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

A. b = (Integer) s;

B. g = (Boolean) s;

C. b = (Integer) g;

D. g = (Boolean) b;

E. Nessuno dei precedenti

---

6. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 z ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc1() );  
            }  
        }  
        static void m()  
            throws Exception {  
                try {  
                }  
                catch( Exception h ) {  
                }  
                finally {  
                    System.out.print(4);  
                }  
            }  
    }
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

11. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

13. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

---

14. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

---

15. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
}
```

```
}

public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

A. Errore a tempo di compilazione
```

- B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 36**

1. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

2. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
```

```

    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new O.I()`.
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {

```

```

public static void main(String [] argv)
    throws Exception {
    try {
        System.out.print(1);
        m();
    }
    catch( Exception s ) {
        System.out.print(2);
        throw( new MyExc1() );
    }
    catch( MyExc2 z ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

12. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
}

```

```

    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

```

```
class Z extends Y {                                }
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
```

- }
- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 37

1. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

4. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 14Exception in thread "main" MyExc1  
 C. 1  
 D. 14222222... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {

```

```

            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3  
 B. 1222222... (ciclo infinito)  
 C. 12  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 7. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {

```

```

        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13  
 B. 1  
 C. 13Exception in thread "main" MyExc2  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

9. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. 6
- 

```

        Integer b;
        Boolean g;
        Object s;
        g = new Boolean(true);
        b = new Integer(50);
        s = new Integer(50);
    
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;  
 B. g = (Boolean) s;  
 C. b = (Integer) g;  
 D. g = (Boolean) b;  
 E. Nessuno dei precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
 B. 101  
 C. 100  
 D. 011  
 E. 010
- 

10. Date le dichiarazioni:

12. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
 B. XYZ  
 C. ZZZ  
 D. Errore a tempo di compilazione.  
 E. Nessuna delle precedenti.
- 

13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
 B. x = y;  
 C. y = m;  
 D. x = m;  
 E. m = (Object []) x;
- 

14. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{

```

```

String type = "Residential";
int age;
public int getAge() {
    return age;
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

15. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
 B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
 C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
 D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
 E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 38

1. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

2. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}
```

```

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 5. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 6. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

#### 7. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    }
}

```

```

        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }

```

```

        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 12222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

11. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
B. Una classe può essere dichiarata abstract o final  
C. Due metodi non possono differire per il tipo di ritorno  
D. Un array non possiede dei membri  
E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {

```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 110  
 B. 101  
 C. 100  
 D. 011  
 E. 010
- 

13. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
 B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
 C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
 D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
 E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
    finally {

```

```

        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}

```

- A. 13  
 B. 1  
 C. 13Exception in thread "main" MyExc2  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();

```

```
Coffee c2 = new Coffee();  
Coffee c3 = new Coffee(40);  
c1 = null;  
c3 = c1;  
/* QUI */  
return;  
}  
}
```

---

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 39**

1. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3

- B. 12222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata `abstract` o `final`
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

7. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}
```

```
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {  
            }  
            catch( MyExc2 j ) {  
            }  
            catch( Exception v ) {  
                System.out.print(2);  
            }  
            finally {  
                throw( new MyExc2() );  
            }  
        }  
        static void m()  
            throws Exception {  
                try {  
                }  
                catch( Exception g ) {  
                }  
                finally {  
                    System.out.print(3);  
                }  
            }  
}
```

- A. 13
- B. 1

C. 13Exception in thread "main" MyExc2

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

9. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 110

B. 101

C. 100

D. 011

E. 010

---

10. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

A. x = (Error []) y;

B. x = y;

C. y = m;

D. x = m;

E. m = (Object []) x;

---

11. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

A. 125

B. 135

C. 1235

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

12. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
    }
}
```

```

    Z z = (Z)y;
    z.f(x); z.f(y); z.f(z);
}
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

13. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

14. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}

```

```

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

16. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 40**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
```

```
}  
catch( MyExc3 y ) {  
    System.out.print(2);  
    throw( new MyExc3() );  
}  
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

4. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0

- D. null
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
}
```

```
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {

```

```

private void f(Y y) {
    System.out.print("Y");
}
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.

```

---

11. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.

```

E. Nessuna delle precedenti.

---

12. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

14. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;
```

```
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}  
  
A. 110  
B. 101  
C. 100  
D. 011  
E. 010
```

---

15. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

16. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal {  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}
```

```
}

}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 41

1. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

2. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

3. Qual è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}
```

```
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

4. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
}  
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

6. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

7. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

9. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
- B. 135

C. 1235

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
```

```

        System.out.print(2);
        throw( new MyExc1() );
    }
    catch( MyExc2 z ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
    try {
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
B. Una classe può essere dichiarata abstract o final  
C. Due metodi non possono differire per il tipo di ritorno  
D. Un array non possiede dei membri  
E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);

```

```

        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

15. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 42

1. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata **abstract** o **final**
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml) {  
        this.ml = ml;  
    }  
    public static void main(String args[]) {  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

3. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}
```

```
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX

- B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
    }
}
```

```
    Mammal m = new Mammal();
}
}
A. 125
B. 135
C. 1235
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```

---

7. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
    }
}
```

```

        catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 9. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
    }
}

```

```

        catch( MyExc3 b ) {
    }
    catch( MyExc2 j ) {
    }
    catch( Exception v ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 11. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```
    }
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

---

12. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

13. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

---

14. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 43

1. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

3. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.

- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

- E. Nessuna delle precedenti.

4. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
}  
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

6. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {  
            }  
            catch( MyExc2 j ) {  
            }  
            catch( Exception v ) {  
                System.out.print(2);  
            }  
            finally {  
                throw( new MyExc2() );  
            }  
    }  
    static void m()  
        throws Exception {  
            try {  
            }  
            catch( Exception h ) {  
            }  
            finally {  
                System.out.print(4);  
            }  
        }  
}
```

```
        catch( Exception g ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 z ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc1() );  
            }  
        }  
        static void m()  
            throws Exception {  
                try {  
                }  
                catch( Exception h ) {  
                }  
                finally {  
                    System.out.print(4);  
                }  
            }  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

13. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal {  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
    }  
}
```

```
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

16. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building {  
    String type = "Residential";
```

```
int age;
public int getAge() {
    return age;
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 44**

1. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

2. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

3. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. Errore a tempo di compilazione

B. 14Exception in thread "main" MyExc1

C. 1

D. 14222222... (ciclo infinito)

E. Nessuna delle precedenti

---

5. Qual è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
}
```

```
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

6. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

A. 125

B. 135

C. 1235

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `x = (Error []) y;`
  - B. `x = y;`
  - C. `y = m;`
  - D. `x = m;`
  - E. `m = (Object []) x;`
- 

8. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (Integer) s;`
  - B. `g = (Boolean) s;`
  - C. `b = (Integer) g;`
  - D. `g = (Boolean) b;`
  - E. Nessuno dei precedenti
- 

11. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**14. Qual è l'output di questo codice?**

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**15. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Dire quale delle seguenti affermazioni è vera:

---

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata **abstract** o **final**
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 45**

1. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
```

```
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

#### 4. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 5. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    }
}

```

```

    /* QUI */
    return;
}
}
A. 1
B. 2
C. 3
D. 4
E. 6

```

---

#### 6. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    }
}

```

```

    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 8. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 9. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

#### 10. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

#### 11. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

12. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {}  
            catch( MyExc2 j ) {}  
            catch( Exception v ) {  
                System.out.print(2);  
            }  
            finally {  
                throw( new MyExc2() );  
            }  
        }  
        static void m()  
    }
```

```
    throws Exception {  
        try {  
        }  
        catch( Exception g ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

15. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new O.I()`.
  - E. Nessuna delle precedenti.
- 

16. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata `abstract` o `final`
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 46**

1. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    finally {

```

```

        throw( new MyExc1() );
    }
}

static void m()
throws Exception {
    try {
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 12222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

#### 7. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```
class Test {
```

#### 6. Quale output si ottiene invocando il metodo p?

```

public static void main(String[] args) {
    X x = new Z();
    Y y = (Y)x;
    Z z = (Z)y;
    z.f(x); z.f(y); z.f(z);
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

8. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 
13. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6
- 

14. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

```

```

    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

15. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
}
```

```
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 47**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

4. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

5. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

6. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

7. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

8. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
        }  
}
```

```

        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 
13. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6
- 
14. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

---

15. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

A. XXX  
B. XYZ  
C. ZZZ

- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata `abstract` o `final`

- C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 48**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
```

```
}  
catch( MyExc3 y ) {  
    System.out.print(2);  
    throw( new MyExc3() );  
}  
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

5. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
```

```
throws Exception {
    try {
        System.out.print(1);
        m();
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc2 j ) {
    }
    catch( Exception v ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
    try {
    }
    catch( Exception g ) {
    }
    finally {
        System.out.print(3);
    }
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

8. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

9. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
```

```
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
```

```

Boolean b1 = new Boolean(true);
if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

13. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

14. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

```

```

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

15. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 49**

1. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
} }
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
}

public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null

E. Nessuna delle precedenti

---

4. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
          Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

5. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

6. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

7. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
```

```

}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3

D. 4

E. 6

---

15. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

A. 125
B. 135
C. 1235
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {

```

```
}

catch( Exception h ) {
}

finally {
    System.out.print(4);
}

}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 50**

1. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

A. 1
B. 2
C. 3
D. 4
E. 6
```

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

```

    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

6. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 9. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);

```

```

                m();
            }
            catch( Exception s ) {
                System.out.print(2);
                throw( new MyExc1() );
            }
            catch( MyExc2 z ) {
                System.out.print(3);
            }
            finally {
                throw( new MyExc1() );
            }
        }
        static void m()
            throws Exception {
            try {
            }
            catch( Exception h ) {
            }
            finally {
                System.out.print(4);
            }
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 11. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

#### 12. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

**13. Qual'è l'output di questo programma?**

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**14. Qual è l'output di questo codice?**

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";

```

```

    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**15. Qual è l'output di questo codice?**

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**16. Quale output si ottiene invocando il metodo p?**

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        }
    }
}

```

```
} else {
    System.out.print(0);
}
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 51**

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

```
System.out.print(1);
Mammal m = new Mammal();
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }

public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }

    static void m()
        throws Exception {

```

```

        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

6. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }

public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }

    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {

```

```

        System.out.print(2);
        throw( new MyExc3() );
    }
}

```

- A. 12Exception in thread "main" MyExc3  
 B. 1222222... (ciclo infinito)  
 C. 12  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
 B. x = y;  
 C. y = m;  
 D. x = m;  
 E. m = (Object []) x;
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
 B. Una classe può essere dichiarata abstract o final  
 C. Due metodi non possono differire per il tipo di ritorno  
 D. Un array non possiede dei membri  
 E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
    }
}

```

```

        catch( Exception v ) {
            System.out.print(2);
        }
    finally {
        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}

```

- A. 13  
 B. 1  
 C. 13Exception in thread "main" MyExc2  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

**13. Date le dichiarazioni:**

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

**14. Qual'è l'output di questo programma?**

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

**15. Qual è l'output di questo codice?**

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
}
```

```
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 52**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
```

```
static Integer ml = 20;

Coffee() {}
Coffee(int ml) {
    this.ml = ml;
}
public static void main(String args[]) {
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}

A. 1
B. 2
C. 3
D. 4
E. 6
```

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

```

static void m()
    throws Exception {
    try {
    }
    catch( Exception g ) {
    }
    finally {
        System.out.print(3);
    }
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

```

    }
}

class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 4. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 6. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010

#### 5. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}

```

---

7. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

8. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

9. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
- 

C. 0

D. null

E. Nessuna delle precedenti

---

10. Qual è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

11. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

12. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
        }
    }
}
```

```
        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 53**

1. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    finally {
        throw( new MyExc1() );
    }
}
```

```
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

4. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
```

```

int age;
public int getAge() {
    return age;
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {

```

```

            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]) {
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}
```

```
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}
```

```
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

#### 9. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}
```

```
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}
```

```
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}
```

```
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;
```

```
        z.f(x); z.f(y); z.f(z);  
    }  
}  
  
A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
```

---

#### 10. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

#### 11. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
    }  
}
```

```

        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
 B. 1222222... (ciclo infinito)  
 C. 12  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
 B. x = y;  
 C. y = m;  
 D. x = m;  
 E. m = (Object []) x;
- 

14. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

16. Dire quale delle seguenti affermazioni è vera:

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 54**

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand = "Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml) {
        this.ml = ml;
    }
    public static void main(String args[]) {
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
```

```
        System.out.print(3);
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 5. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

#### 6. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 8. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null

E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. Errore a tempo di compilazione

B. 14Exception in thread "main" MyExc1

C. 1

D. 14222222... (ciclo infinito)

E. Nessuna delle precedenti

10. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.

D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

E. Nessuna delle precedenti.

11. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 110

B. 101

C. 100

D. 011

E. 010

12. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (Integer) s;`
- B. `g = (Boolean) s;`
- C. `b = (Integer) g;`
- D. `g = (Boolean) b;`
- E. Nessuno dei precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
```

```
        System.out.println(h.getAge());
    }
}

A. Errore a tempo di compilazione
B. Errore a tempo di esecuzione
C. 0
D. null
E. Nessuna delle precedenti
```

---

15. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

16. Qual è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 55**

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

2. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

3. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

4. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

7. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

10. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);

```

```

            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13  
 B. 1  
 C. 13Exception in thread "main" MyExc2  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 14. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
    }
}

```

```

        Mammal m = new Mammal();
    }
}
A. 125
B. 135
C. 1235
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

#### 15. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
A. 125
B. 135
C. 1235
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

#### 16. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {

```

```
    System.out.print(0);                }
}                                }
if(s3 == s2) {
    System.out.print(1);            A. 110
} else {
    System.out.print(0);            B. 101
}
if(s4 == s5) {
    System.out.print(1);            C. 100
} else {
    System.out.print(0);            D. 011
}
}
```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 56**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
```

```
static Integer ml = 20;

Coffee() {}
Coffee(int ml) {
    this.ml = ml;
}
public static void main(String args[]) {
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null

E. Nessuna delle precedenti

4. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.

5. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
    }
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception q ) {

```

```

    }
    finally {
        System.out.print(3);
    }
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;
- 

9. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione
- 

E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
-

12. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

15. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

16. Qual è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}
```

```
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- ```
    }  
  
A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
```
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 57**

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

2. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

5. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

6. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

---

E. Nessuna delle precedenti.

---

7. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( Exception s ) {  
            System.out.print(2);  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 z ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc1() );  
        }  
    }  
    static void m()  
        throws Exception {
```

```

try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
}

```

```

        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

12. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {
```

```
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

15. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal {  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-

16. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono

corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 58**

1. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
-

4. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {
```

```
public static void main(String [] argv)  
throws Exception {  
    try {  
        System.out.print(1);  
        m();  
    }  
    catch( Exception s ) {  
        System.out.print(2);  
        throw( new MyExc1() );  
    }  
    catch( MyExc2 z ) {  
        System.out.print(3);  
    }  
    finally {  
        throw( new MyExc1() );  
    }  
}  
static void m()  
throws Exception {  
    try {  
    }  
    catch( Exception h ) {  
    }  
    finally {  
        System.out.print(4);  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

8. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;

D. `x = m;`  
E. `m = (Object []) x;`

---

9. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
```

```
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

14. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 59**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
} }
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

5. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
}
```

```
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata **abstract** o **final**
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

- C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 11. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

#### 12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- B. 101
  - C. 100
  - D. 011
  - E. 010
- 

13. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {                                }
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
```

- }
- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 60**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
}
```

```
}  
Animal(int i) {  
    System.out.print(3);  
}  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }
}  
A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
```

---

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }
}  
public static void main(String args[]){  
    Coffee c1 = new Coffee();  
    Coffee c2 = new Coffee();  
    Coffee c3 = new Coffee(40);  
    c1 = null;  
    c3 = c1;  
    /* QUI */  
    return;  
}
```

- A. 1
- B. 2

C. 3

D. 4

E. 6

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

A. 12Exception in thread "main" MyExc3

B. 1222222... (ciclo infinito)

C. 12

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

5. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
```

```
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

6. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. 0

D. null

E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

A. b = (Integer) s;

B. g = (Boolean) s;

C. b = (Integer) g;

D. g = (Boolean) b;

E. Nessuno dei precedenti

---

8. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

11. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

12. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;

E. `m = (Object []) x;`

---

13. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

16. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}
```

```
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 61**

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
```

```

public static void main(String args[]) {
    System.out.print(1);
    Mammal m = new Mammal();
}

```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    }
}

```

```

        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 14Exception in thread "main" MyExc1  
 C. 1  
 D. 14222222... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

7. Qual è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
 B. XYZ  
 C. ZZZ  
 D. Errore a tempo di compilazione.  
 E. Nessuna delle precedenti.
- 

8. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

9. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }
```

```
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

---

12. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        } catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        } catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;

- D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

15. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
    }
}

```

```
        }
    finally {
        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
        }
    }
}
A. 13
B. 1
C. 13Exception in thread "main" MyExc2
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 62**

1. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
```

```
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
```

```

    }
}

class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
- B. 1

C. 13Exception in thread "main" MyExc2

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 7. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}

class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}

class Rabbit extends Mammal {

```

```

public static void main(String args[]) {
    System.out.print(1);
    Mammal m = new Mammal();
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse

- B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
        static void p() {
            try {
                System.out.print(1);
                throw( new MyExc3() );
            }
            catch( MyExc3 y ) {
                System.out.print(2);
                throw( new MyExc3() );
            }
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 12222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    }
}

```

```

    /* QUI */
    return;
}
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

13. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

15. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

16. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {

```

```
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- 
- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 63

1. Qual è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
}
```

```
}  
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

4. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

5. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 7. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }

    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2

- C. 3
- D. 4
- E. 6

11. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

A. 125
B. 135
C. 1235
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

12. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 14Exception in thread "main" MyExc1  
 C. 1  
 D. 14222222... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

#### 15. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
 B. 12222222... (ciclo infinito)  
 C. 12  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 16. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {

```

```
System.out.print(1);  
} else {  
    System.out.print(0);  
}  
  
if(s3 == s2) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}  
  
if(s4 == s5) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 64**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
```

```
public static void main(String args[]) {
    System.out.print(1);
    Mammal m = new Mammal();
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. 0

D. null

E. Nessuna delle precedenti

---

#### 6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
B. Una classe può essere dichiarata abstract o final  
C. Due metodi non possono differire per il tipo di ritorno  
D. Un array non possiede dei membri  
E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

#### 7. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

#### 8. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

10. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

11. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
```

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

12. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

13. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
        }  
}
```

```
        catch( MyExc2 z ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc1() );  
        }  
    }  
    static void m()  
        throws Exception {  
            try {  
            }  
            catch( Exception h ) {  
            }  
            finally {  
                System.out.print(4);  
            }  
        }  
}
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {                                     /* QUI */
    String brand ="Nespresso";
    static Integer ml = 20;
}
Coffee() {}
Coffee(int ml){
    this.ml = ml;
}
public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
```

---

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 65

1. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
}
```

```
}  
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

4. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}
```

```

}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

6. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

```

    }
}

A. 13
B. 1
C. 13Exception in thread "main" MyExc2
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

8. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

11. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

**14. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {

```

```

        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

**15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?**

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

**16. Qual è l'output di questo codice?**

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
}

```

```
static void p() {  
    try {  
        System.out.print(1);  
        throw( new MyExc3() );  
    }  
    catch( MyExc3 y ) {  
        System.out.print(2);  
        throw( new MyExc3() );  
    }  
}
```

```
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 66**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
```

```
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[])
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null

E. Nessuna delle precedenti

---

4. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
    }
```

```
        System.out.print(2);  
        throw( new MyExc3() );  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( Exception s ) {  
            System.out.print(2);  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 z ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc1() );  
        }  
    }  
    static void m()  
        throws Exception {  
        try {  
        }  
        catch( Exception h ) {  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

---

7. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 8. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

#### 9. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

#### 10. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
    }
}

```

```

        c1 = null;
        c3 = c1;
    /* QUI */
    return;
}
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

16. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 67**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
```

```
public static void main(String args[]) {
    System.out.print(1);
    Mammal m = new Mammal();
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    }
}
```

```

        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

6. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {

```

```

        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

7. Qual è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {}  
            catch( MyExc2 j ) {}  
            catch( Exception v ) {  
                System.out.print(2);  
            }  
            finally {  
                throw( new MyExc2() );  
            }  
        }  
        static void m()  
            throws Exception {
```

```
        try {  
        }  
        catch( Exception g ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;
```

```
/* QUI */  
    return;  
}  
}  
  
A. 1  
B. 2  
C. 3  
D. 4  
E. 6
```

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 68**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}
```

```
} }
```

```
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}
```

```
class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
}
```

```
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

```

```

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

6. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

---

7. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

8. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final

- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

---

10. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

---

11. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

14. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}
```

```
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

16. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
        }
```

```
        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 69**

1. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

2. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

3. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
```

```

        System.out.print(3);
    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
    try {
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
```

```

        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 12222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

9. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

A. 110
B. 101
C. 100
D. 011
E. 010

```

---

10. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;

E. `m = (Object []) x;`

---

12. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata `abstract` o `final`
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 70**

1. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
```

```
} catch( MyExc3 b ) {
} catch( MyExc2 j ) {
} catch( Exception v ) {
    System.out.print(2);
}
finally {
    throw( new MyExc2() );
}
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.

D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

E. Nessuna delle precedenti.

---

4. Qual è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

5. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

A. b = (Integer) s;

B. g = (Boolean) s;

C. b = (Integer) g;

D. g = (Boolean) b;

E. Nessuno dei precedenti

---

6. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. 0

D. null

E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. 0

D. null

E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

#### 9. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101

C. 100

D. 011

E. 010

---

#### 10. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;
- 

#### 11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)

E. Nessuna delle precedenti

---

12. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

A. 125

B. 135

C. 1235

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata `abstract` o `final`
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

---

14. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
    }  
}
```

```
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

A. 125

B. 135

C. 1235

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}  
  
A. 1  
B. 2  
C. 3  
D. 4  
E. 6
```

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
}
```

```
static void p() {  
    try {  
        System.out.print(1);  
        throw( new MyExc3() );  
    }  
    catch( MyExc3 y ) {  
        System.out.print(2);  
        throw( new MyExc3() );  
    }  
}
```

```
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 71**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
} }
```

```
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.

- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

- E. Nessuna delle precedenti.

---

4. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null

- E. Nessuna delle precedenti
- 

6. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

7. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;
- 

8. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;
```

```

public int getAge() {
    return age;
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }

```

```

public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

13. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
    }  
}  
  
A. 110  
B. 101  
C. 100  
D. 011  
E. 010
```

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 72**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;

- D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
```

```

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

6. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

    }

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
 B. XYZ  
 C. ZZZ  
 D. Errore a tempo di compilazione.  
 E. Nessuna delle precedenti.
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
 B. Una classe può essere dichiarata **abstract** o **final**  
 C. Due metodi non possono differire per il tipo di ritorno  
 D. Un array non possiede dei membri  
 E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {

```

```

        }
    finally {
        System.out.print(3);
    }
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;
- 

10. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        }
    }
}

```

```

    } else {
        System.out.print(0);
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

13. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";

```

```

    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

```

```
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.

---

**Prova n. 1**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 2**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 3**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 4**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 5**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 6**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 8**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 9**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 10**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 11**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 12**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 13**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 15**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 16**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 1:

1			■				
2				■	■		
3					■	■	
4	■	■					
5			■	■			
6	■	■					
7				■	■		
8					■	■	
9	■	■					
10						■	■
11			■	■			
12					■	■	
13						■	■
14					■	■	
15						■	■
16	■	■					

Risultato prova n. 2:

1			■				
2				■	■		
3			■	■			
4	■	■					
5					■	■	
6					■	■	
7	■	■					
8						■	■
9				■	■		
10	■	■					
11	■	■					
12					■	■	
13				■	■		
14			■	■			
15						■	■
16				■	■		

Risultato prova n. 3:

1		■					
2							■
3				■	■		
4				■	■		
5			■				
6				■	■		
7				■	■		
8			■				
9				■	■		
10				■	■		
11		■	■				
12		■	■				
13		■	■				
14			■				■
15			■				
16					■	■	

Risultato prova n. 4:

A 16x16 grid containing red squares at the following coordinates:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (1, 5)
- (2, 5)
- (3, 5)
- (4, 5)
- (5, 5)
- (6, 5)
- (7, 5)
- (8, 5)
- (9, 5)
- (10, 5)
- (11, 5)
- (12, 5)
- (13, 5)
- (14, 5)
- (15, 5)
- (16, 5)
- (1, 6)
- (2, 6)
- (3, 6)
- (4, 6)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (1, 7)
- (2, 7)
- (3, 7)
- (4, 7)
- (5, 7)
- (6, 7)
- (7, 7)
- (8, 7)
- (9, 7)
- (10, 7)
- (11, 7)
- (12, 7)
- (13, 7)
- (14, 7)
- (15, 7)
- (16, 7)
- (1, 8)
- (2, 8)
- (3, 8)
- (4, 8)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (1, 9)
- (2, 9)
- (3, 9)
- (4, 9)
- (5, 9)
- (6, 9)
- (7, 9)
- (8, 9)
- (9, 9)
- (10, 9)
- (11, 9)
- (12, 9)
- (13, 9)
- (14, 9)
- (15, 9)
- (16, 9)
- (1, 10)
- (2, 10)
- (3, 10)
- (4, 10)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (1, 11)
- (2, 11)
- (3, 11)
- (4, 11)
- (5, 11)
- (6, 11)
- (7, 11)
- (8, 11)
- (9, 11)
- (10, 11)
- (11, 11)
- (12, 11)
- (13, 11)
- (14, 11)
- (15, 11)
- (16, 11)
- (1, 12)
- (2, 12)
- (3, 12)
- (4, 12)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (1, 13)
- (2, 13)
- (3, 13)
- (4, 13)
- (5, 13)
- (6, 13)
- (7, 13)
- (8, 13)
- (9, 13)
- (10, 13)
- (11, 13)
- (12, 13)
- (13, 13)
- (14, 13)
- (15, 13)
- (16, 13)
- (1, 14)
- (2, 14)
- (3, 14)
- (4, 14)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)
- (1, 15)
- (2, 15)
- (3, 15)
- (4, 15)
- (5, 15)
- (6, 15)
- (7, 15)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 15)
- (16, 15)
- (1, 16)
- (2, 16)
- (3, 16)
- (4, 16)
- (5, 16)
- (6, 16)
- (7, 16)
- (8, 16)
- (9, 16)
- (10, 16)
- (11, 16)
- (12, 16)
- (13, 16)
- (14, 16)
- (15, 16)
- (16, 16)

## Risultato prova n. 5:

1					
2					
3			■		
4					■
5					■
6	■				
7				■	
8					■
9					■
10	■				
11	■				
12				■	
13	■				
14			■		
15					■
16		■			

Risultato prova n. 6:

1		■					
2							■
3					■	■	
4							■
5			■				
6				■	■		
7				■	■		
8		■					
9						■	
10				■	■		
11		■					
12			■				
13		■					
14			■				
15				■	■		
16						■	

Risultato prova n. 7:

1						
2						
3		■				
4					■	
5			■			
6		■				
7		■				
8						■
9				■		
10				■		
11						■
12				■		
13			■			
14				■		
15			■			
16	■					

Risultato prova n. 8:

1		■					
2				■			
3		■					
4					■		
5						■	
6				■			
7						■	
8				■			
9			■				
10		■					
11			■				
12					■		
13		■					
14					■		
15			■				
16				■			

Risultato prova n. 9:

1		■					
2						■	
3					■	■	
4			■				
5						■	
6				■	■		
7				■	■		
8			■	■			
9			■	■			
10				■	■		
11					■	■	
12					■	■	
13		■	■				
14				■	■		
15		■	■				
16		■	■				

Risultato prova n. 10:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 11:

1			■					
2				■				
3					■			
4						■		
5							■	
6								■
7								■
8		■						
9			■					
10								■
11						■		
12					■			
13				■				
14		■						
15			■					
16								■

Risultato prova n. 12:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 13:

1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Risultato prova n. 14:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 15:

1		■					
2			■				
3		■					
4						■	
5			■				
6			■				
7				■			
8					■		
9				■			
10		■					
11		■					
12			■				
13					■		
14					■		
15					■		
16			■				

Risultato prova n. 16:

1		■					
2				■	■		
3						■	■
4			■	■			
5					■	■	
6							■
7						■	■
8		■	■				
9					■	■	
10		■	■				
11					■	■	
12						■	■
13			■	■			
14			■	■			
15		■	■				
16							■

Risultato prova n. 17:

1		■					
2			■				
3				■			
4				■			
5					■		
6		■					
7		■					
8			■				
9				■			
10				■			
11					■		
12		■					
13				■			
14					■		
15						■	
16						■	

Risultato prova n. 18:

1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	

Risultato prova n. 19:

1			■				
2					■	■	
3						■	■
4				■			
5					■	■	
6	■	■					
7					■	■	
8			■	■			
9			■	■			
10	■	■					
11					■		
12	■	■					
13	■	■					
14				■	■		
15				■	■		
16						■	

Risultato prova n. 20:

1		■					
2				■	■		
3			■	■			
4					■	■	
5		■	■				
6			■	■			
7			■	■			
8					■	■	
9						■	■
10					■	■	
11						■	■
12	■	■					
13							■
14	■	■					
15					■	■	
16							■

Risultato prova n. 21:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 22:

1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Risultato prova n. 23:

1			■				
2	■						
3					■	■	
4			■	■			
5			■	■			
6	■	■					
7	■	■					
8	■						
9					■	■	
10				■	■		
11				■	■		
12				■	■		
13				■	■		
14						■	■
15						■	■
16					■		

Risultato prova n. 24:

1			■				
2					■	■	
3					■	■	
4				■			
5						■	
6	■	■					
7			■	■			
8	■	■					
9				■	■		
10			■	■			
11					■	■	
12				■	■		
13				■	■		
14						■	
15	■	■					
16	■	■					

Risultato prova n. 25:

1		■					
2			■				
3		■					
4				■			
5			■				
6					■		
7		■					
8				■			
9		■					
10					■		
11						■	
12				■			
13						■	
14					■		
15			■				
16				■			

Risultato prova n. 26:

1		■					
2				■			
3							■
4					■		
5						■	
6		■					
7						■	
8			■				
9				■			
10		■					
11			■				
12				■			
13						■	
14				■			
15					■		
16		■					

Risultato prova n. 27:

1		■					
2			■				
3		■					
4				■			
5					■		
6					■		
7		■					
8		■					
9				■			
10			■				
11				■			
12			■				
13					■		
14					■		
15					■		
16			■				

Risultato prova n. 28:

1					
2	■				
3		■			
4					■
5			■		
6		■			
7		■			
8				■	
9	■				
10				■	
11			■		
12			■		
13	■				
14					■
15				■	
16	■				

Risultato prova n. 29:

1		■					
2					■	■	
3			■	■			
4					■	■	
5			■	■			
6					■	■	
7						■	■
8					■	■	
9						■	■
10		■	■				
11			■	■			
12	■	■					
13						■	■
14				■	■		
15	■	■					
16					■	■	

Risultato prova n. 30:

1		■					
2			■				
3						■	
4				■			
5						■	
6	■						
7				■			
8				■			
9				■			
10	■						
11					■		
12		■					
13	■						
14					■		
15			■				
16					■		

Risultato prova n. 31:

## Risultato prova n. 32:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 33:

A 16x16 grid with red squares at the following coordinates:

- (5, 2)
- (5, 3)
- (6, 2)
- (6, 3)
- (8, 2)
- (8, 3)
- (9, 4)
- (9, 5)
- (10, 4)
- (10, 5)
- (11, 4)
- (11, 5)
- (12, 4)
- (12, 5)
- (13, 4)
- (13, 5)
- (14, 4)
- (14, 5)
- (15, 4)
- (15, 5)
- (16, 2)
- (16, 3)

## Risultato prova n. 34:

1			■				
2				■	■		
3				■	■		
4						■	■
5	■	■					
6						■	■
7	■	■					
8	■						
9					■	■	
10			■	■			
11				■	■		
12				■	■		
13					■	■	
14			■	■			
15					■	■	
16	■	■					

Risultato prova n. 35:

1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Risultato prova n. 36:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 37:

1							1
2							2
3							3
4							4
5							5
6							6
7							7
8							8
9							9
10							10
11							11
12							12
13							13
14							14
15							15
16							16

Risultato prova n. 38:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 39:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 40:

A 16x16 grid with red squares at specific intersections. The red squares are located at the following coordinates (row, column): (1, 16), (2, 3), (3, 16), (4, 2), (4, 16), (5, 2), (5, 16), (6, 4), (7, 2), (8, 4), (9, 4), (10, 3), (11, 4), (12, 2), (13, 3), (14, 4), (15, 4), and (16, 2).

## Risultato prova n. 41:

1			■				
2				■	■		
3				■	■		
4						■	■
5				■	■		
6			■	■			
7	■	■					
8	■						
9					■	■	
10				■	■		
11					■	■	
12						■	■
13			■	■			
14	■	■					
15						■	■
16	■	■					

Risultato prova n. 42:

1							1
2							2
3							3
4							4
5							5
6							6
7							7
8							8
9							9
10							10
11							11
12							12
13							13
14							14
15							15
16							16

Risultato prova n. 43:

1			■		
2				■	
3					■
4	■				
5				■	
6		■			
7					■
8				■	
9			■		
10	■				
11	■				
12			■		
13			■		
14	■				
15			■		
16		■			

Risultato prova n. 44:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 45:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 46:

1			■		
2		■			
3			■		
4				■	
5	■				
6					■
7				■	
8		■			
9			■		
10	■				
11	■				
12	■				
13			■		
14				■	
15					■
16		■			

Risultato prova n. 47:

1			■				
2				■			
3					■		
4						■	
5						■	
6				■			
7	■						
8			■				
9	■						
10	■						
11				■			
12					■		
13			■				
14						■	
15				■			
16	■						

Risultato prova n. 48:

A 16x16 grid with red squares at the following coordinates:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (1, 5)
- (2, 5)
- (3, 5)
- (4, 5)
- (5, 5)
- (6, 5)
- (7, 5)
- (8, 5)
- (9, 5)
- (10, 5)
- (11, 5)
- (12, 5)
- (13, 5)
- (14, 5)
- (15, 5)
- (16, 5)
- (1, 6)
- (2, 6)
- (3, 6)
- (4, 6)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (1, 7)
- (2, 7)
- (3, 7)
- (4, 7)
- (5, 7)
- (6, 7)
- (7, 7)
- (8, 7)
- (9, 7)
- (10, 7)
- (11, 7)
- (12, 7)
- (13, 7)
- (14, 7)
- (15, 7)
- (16, 7)
- (1, 8)
- (2, 8)
- (3, 8)
- (4, 8)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (1, 9)
- (2, 9)
- (3, 9)
- (4, 9)
- (5, 9)
- (6, 9)
- (7, 9)
- (8, 9)
- (9, 9)
- (10, 9)
- (11, 9)
- (12, 9)
- (13, 9)
- (14, 9)
- (15, 9)
- (16, 9)
- (1, 10)
- (2, 10)
- (3, 10)
- (4, 10)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (1, 11)
- (2, 11)
- (3, 11)
- (4, 11)
- (5, 11)
- (6, 11)
- (7, 11)
- (8, 11)
- (9, 11)
- (10, 11)
- (11, 11)
- (12, 11)
- (13, 11)
- (14, 11)
- (15, 11)
- (16, 11)
- (1, 12)
- (2, 12)
- (3, 12)
- (4, 12)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (1, 13)
- (2, 13)
- (3, 13)
- (4, 13)
- (5, 13)
- (6, 13)
- (7, 13)
- (8, 13)
- (9, 13)
- (10, 13)
- (11, 13)
- (12, 13)
- (13, 13)
- (14, 13)
- (15, 13)
- (16, 13)
- (1, 14)
- (2, 14)
- (3, 14)
- (4, 14)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)
- (1, 15)
- (2, 15)
- (3, 15)
- (4, 15)
- (5, 15)
- (6, 15)
- (7, 15)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 15)
- (16, 15)
- (1, 16)
- (2, 16)
- (3, 16)
- (4, 16)
- (5, 16)
- (6, 16)
- (7, 16)
- (8, 16)
- (9, 16)
- (10, 16)
- (11, 16)
- (12, 16)
- (13, 16)
- (14, 16)
- (15, 16)
- (16, 16)

## Risultato prova n. 49:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 50:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 51:

1		■					
2				■	■		
3				■	■		
4						■	■
5			■	■			
6					■	■	
7							■
8		■	■				
9		■	■				
10					■	■	
11				■	■		
12			■	■			
13			■	■			
14				■	■		
15							■
16	■	■					

Risultato prova n. 52:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 53:

1					
2	■				
3		■			
4			■		
5					■
6				■	■
7				■	■
8			■		
9	■				
10		■			
11				■	
12	■				
13			■		
14	■				
15		■			
16					■

Risultato prova n. 54:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 55:

1		■					
2				■			
3		■					
4						■	
5					■		
6			■				
7				■			
8						■	
9					■		
10				■			
11			■				
12	■						
13				■			
14	■						
15			■				
16					■		

Risultato prova n. 56:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 57:

1		■					
2							■
3					■	■	
4		■					
5				■	■		
6		■					
7			■				
8							■
9				■	■		
10						■	
11				■	■		
12			■	■			
13				■	■		
14		■	■				
15						■	
16			■				

Risultato prova n. 58:

1			■				
2						■	■
3					■	■	
4						■	■
5	■	■					
6	■	■					
7			■	■			
8				■	■		
9	■	■					
10				■	■		
11	■	■					
12					■	■	
13			■	■			
14				■	■		
15				■	■		
16					■	■	

Risultato prova n. 59:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 60:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 61:

## Risultato prova n. 62:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 63:

1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Risultato prova n. 64:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 65:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 66:

1			■				
2					■	■	
3						■	■
4	■	■					
5	■	■					
6					■	■	
7						■	■
8				■	■		
9	■	■					
10				■	■		
11					■	■	
12	■	■					
13				■	■		
14			■	■			
15				■	■		
16		■	■				

Risultato prova n. 67:

1		■					
2						■	
3				■	■		
4					■		
5						■	■
6						■	■
7		■	■				
8		■					
9			■	■			
10			■	■			
11						■	
12		■	■				
13							■
14			■	■			
15					■		
16					■		

Risultato prova n. 68:

1							1
2							2
3							3
4							4
5							5
6							6
7							7
8							8
9							9
10							10
11							11
12							12
13							13
14							14
15							15
16							16

Risultato prova n. 69:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 70:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 71:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 72:



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 1

1. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(){  
    }  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

- 
3. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args){  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

4. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}  
  
A. 011  
B. 111  
C. 110  
D. 010  
E. 100
```

---

**5. Dato il seguente codice:**

```
1. public interface I {  
2.     int n = 0;  
3. }
```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

---

**6. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        static void m() {  
            try {  
                System.out.print(4);  
                catch( MyExc2 f ) {  
                    System.out.print(5);  
                }  
                catch( MyExc3 v ) {  
                    System.out.print(6);  
                    throw( new MyExc1() );  
                }  
            }  
        }  
    }  
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356
- E. Nessuna delle precedenti

---

**7. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

---

## 8. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

---

## 9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
        }  
        catch( MyExc1 v ) {  
            System.out.print(2);  
        }  
        catch( Exception d ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc2() );  
        }  
    }  
    static void q()  
        throws Exception {  
        try {  
            System.out.print(5);  
            if (true) throw( new MyExc3() );  
            else throw( new MyExc1() );  
        }  
        catch( MyExc1 z ) {  
            System.out.print(6);  
        }  
        finally {  
            System.out.print(7);  
        }  
    }  
}
```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

**10. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.**
- B. Una classe interna non può essere dichiarata final.**
- C. Una classe interna non può essere dichiarata static.**
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.**
- E. Nessuna delle precedenti**

---

**11. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145**
- B. 145632**
- C. Errore a tempo di compilazione**
- D. 145623Exception in thread "main" MyExc2**
- E. Nessuna delle precedenti**

---

**12. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}   
  
    public static String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat", "500");  
        Vehicle c = new Car("Opel", "Corsa");  
        Car d = new Car("Citroen", "C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3**
- B. Fiat Opel Corsa Citroen C3**
- C. Citroen Citroen Citroen C3**
- D. Errore a tempo di compilazione.**
- E. Errore a tempo di esecuzione.**

---

**13. Date le dichiarazioni:**

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;**
- B. q = (Error) m;**
- C. m = (Boolean) q;**
- D. m = (Boolean) u;**
- E. Nessuno dei precedenti**

---

**14. Date le dichiarazioni:**

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

**e le dichiarazioni di variabile:**

```
Super n;  
B1 r;  
C2 z;
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;**
- B. n = r;**
- C. z = r;**
- D. r = z;**
- E. z = n;**

---

**15. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}   
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

**16. Quale output si ottiene invocando il metodo g?**

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
}  
void g() {  
    Y y = new Y();  
    Object o = new Z();  
    f(y); f(o);  
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 2**

1. Dato il seguente codice:

```
1. public interface I {
2.     int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}
```

```
class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
    }
}
```

```
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}
```

```

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
 B. Fiat Opel Corsa Citroen C3  
 C. Citroen Citroen Citroen C3  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
    }
}

```

```

        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
 B. Fiat Opel Corsa Citroen C3  
 C. Citroen Citroen Citroen C3  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

6. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}

class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}

void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX  
 B. X Z  
 C. ZZ  
 D. Errore a tempo di compilazione  
 E. Errore a tempo di esecuzione
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata **private**.  
 B. Una classe interna non può essere dichiarata **final**.  
 C. Una classe interna non può essere dichiarata **static**.  
 D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.  
 E. Nessuna delle precedenti
- 

8. Quanti oggetti **al massimo possono essere deallocati al punto indicato?**

```

        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 1**  
**B. 2**  
**C. 3**  
**D. 4**  
**E. Nessuna delle precedenti**
- 

**9. Data la seguente classe:**

```

1. public class C {
2.     public void f( int i){}
3.
4. }
```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)**  
**B. public int f( int j)**  
**C. private void f( int j)**  
**D. public static void f( int j)**  
**E. Nessuna delle precedenti**
- 

**10. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

**11. Date le dichiarazioni:**

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;**  
**B. q = (Error) m;**  
**C. m = (Boolean) q;**  
**D. m = (Boolean) u;**  
**E. Nessuno dei precedenti**
- 

**12. Qual'è l'output di questo programma?**

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.**

**B. Viene lanciata una**

NullPointerException.

**C. Viene lanciata una**

ArrayIndexOutOfBoundsException.

**D. null**

**E. Nessuna delle precedenti.**

### 13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

**A. 1574Exception in thread "main" MyExc2**

**B. 157**

**C. 157342**

**D. Errore a tempo di compilazione**

**E. Nessuna delle precedenti**

### 14. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

**A. 011**

**B. 111**

**C. 110**

**D. 010**

**E. 100**

### 15. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri**
- B. Quando due metodi hanno lo stesso nome si ha overloading**
- C. Non tutti i tipi di eccezioni estendono la classe Object**
- D. Un array non possiede dei membri**
- E. Due metodi possono differire per il tipo di ritorno**

### 16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
    }
}

```

```

}
catch( Exception i ) {
    System.out.print(2);
}
catch( MyExc3 a ) {
}
finally {
    System.out.print(3);
    throw( new MyExc2() );
}
}

static void p()
throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}

```

- A. 145**
- B. 145632**
- C. Errore a tempo di compilazione**
- D. 145623Exception in thread "main" MyExc2**
- E. Nessuna delle precedenti**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 3**

1. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

2. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

3. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1**
  - B. 2**
  - C. 3**
  - D. 4**
  - E. Nessuna delle precedenti**
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

```

static void p()
    throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc2 f ) {
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc1 f ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
        throw( new MyExc3() );
    }
}
}

void m(Integer i2, Object f2, Integer i3) {
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 145**  
**B. 145632**  
**C. Errore a tempo di compilazione**  
**D. 145623Exception in thread "main" MyExc2**  
**E. Nessuna delle precedenti**
- 

#### 5. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;**  
**B. n = r;**  
**C. z = r;**  
**D. r = z;**  
**E. z = n;**
- 

#### 6. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
}

```

- A. 011**  
**B. 111**  
**C. 110**  
**D. 010**  
**E. 100**
- 

#### 7. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.**  
**B. Viene lanciata una NullPointerException.**  
**C. Viene lanciata una ArrayIndexOutOfBoundsException.**  
**D. null**  
**E. Nessuna delle precedenti.**
- 

#### 8. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri**  
**B. Quando due metodi hanno lo stesso nome si ha overloading**  
**C. Non tutti i tipi di eccezioni estendono la classe Object**  
**D. Un array non possiede dei membri**  
**E. Due metodi possono differire per il tipo di ritorno**
-

**9. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

**10. Date le dichiarazioni:**

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;

**E. Nessuno dei precedenti**

**11. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3
B. Fiat Opel Corsa Citroen C3
C. Citroen Citroen Citroen C3
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

**12. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {

```

```

    }
}

static void m() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    catch( MyExc2 f ) {
        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

**A. 135**

**B. 1356Exception in thread "main" MyExc1**

**C. Errore a tempo di compilazione**

**D. 1356**

**E. Nessuna delle precedenti**

**13. Dire quale delle seguenti affermazioni è vera:**

**A. Una classe interna non può essere dichiarata private.**

**B. Una classe interna non può essere dichiarata final.**

**C. Una classe interna non può essere dichiarata static.**

**D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.**

**E. Nessuna delle precedenti**

**14. Dato il seguente codice:**

```

1. public interface I {
2.     int n = 0;
3. }

```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

**A. final int n = 0;**

**B. abstract int n = 0;**

**C. protected int n = 0;**

**D. public int n;**

**E. Nessuna delle precedenti**

**15. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

**A. 1574Exception in thread "main" MyExc2**

**B. 157**

**C. 157342**

**D. Errore a tempo di compilazione**

**E. Nessuna delle precedenti**

**16. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

```

```

    }

}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
- B. Fiat Opel Corsa Citroen C3**
- C. Citroen Citroen Citroen C3**
- D. Errore a tempo di compilazione.**
- E. Errore a tempo di esecuzione.**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 4

1. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}  
  
    public static String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

A. Fiat Opel Citroen C3

- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        static void m() {  
            try {  
                System.out.print(4);  
            }  
            catch( MyExc2 f ) {  
                System.out.print(5);  
            }  
        }  
    }  
}
```

```

        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 135  
B. 1356Exception in thread "main" MyExc1  
C. Errore a tempo di compilazione  
D. 1356  
E. Nessuna delle precedenti
- 

**5. Quanti oggetti *al massimo possono essere deallocati al punto indicato?***

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

**6. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.  
B. Una classe interna non può essere dichiarata final.  
C. Una classe interna non può essere dichiarata static.

- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.  
E. Nessuna delle precedenti
- 

**7. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

**8. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {

```

```

        System.out.print(1);
        p();
    }
    catch( Exception i ) {
        System.out.print(2);
    }
    catch( MyExc3 a ) {
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
static void p()
throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}

```

**A. 145**

**B. 145632**

**C. Errore a tempo di compilazione**

**D. 145623Exception in thread "main" MyExc2**

**E. Nessuna delle precedenti**

## 9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}
}

```

```

    public String id() {
        return producer + " " + model + " ";
    }

    class Main {
        public static void main(String[] args) {
            Vehicle v = new Vehicle("Fiat","500");
            Vehicle c = new Car("Opel","Corsa");
            Car d = new Car("Citroen","C3");
            System.out.print(v.id());
            System.out.print(c.id());
            System.out.print(d.id());
        }
    }
}

```

**A. Fiat Opel Citroen C3**

**B. Fiat Opel Corsa Citroen C3**

**C. Citroen Citroen Citroen C3**

**D. Errore a tempo di compilazione.**

**E. Errore a tempo di esecuzione.**

## 10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

**A. Fiat Opel Citroen C3**

- B. Fiat Opel Corsa Citroen C3**
- C. Citroen Citroen Citroen C3**
- D. Errore a tempo di compilazione.**
- E. Errore a tempo di esecuzione.**

---

## 11. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.**
- B. Viene lanciata una NullPointerException.**
- C. Viene lanciata una ArrayIndexOutOfBoundsException.**
- D. null**
- E. Nessuna delle precedenti.**

---

## 12. Date le dichiarazioni:

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

### e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;**
- B. n = r;**
- C. z = r;**
- D. r = z;**
- E. z = n;**

---

## 13. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;**
- B. abstract int n = 0;**
- C. protected int n = 0;**
- D. public int n;**
- E. Nessuna delle precedenti**

---

## 14. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;**
- B. q = (Error) m;**
- C. m = (Boolean) q;**
- D. m = (Boolean) u;**
- E. Nessuno dei precedenti**

---

## 15. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 011**
- B. 111**
- C. 110**
- D. 010**
- E. 100**

---

**16. Quale output si ottiene invocando il metodo g?**

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
}
```

```
Y y = new Y();  
Object o = new Z();  
f(y); f(o);  
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 5**

1. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
}
```

```
public String id() {
    return producer + " " + model + " ";
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
    }
}
```

```

        }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

#### 4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

```

        }
    }

```

- A. 145**  
**B. 145632**  
**C. Errore a tempo di compilazione**  
**D. 145623Exception in thread "main" MyExc2**  
**E. Nessuna delle precedenti**
- 

#### 5. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri**  
**B. Quando due metodi hanno lo stesso nome si ha overloading**  
**C. Non tutti i tipi di eccezioni estendono la classe Object**  
**D. Un array non possiede dei membri**  
**E. Due metodi possono differire per il tipo di ritorno**
- 

#### 6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**

- B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

**7. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

**8. Dato il seguente codice:**

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;**
  - B. abstract int n = 0;**
  - C. protected int n = 0;**
  - D. public int n;**
  - E. Nessuna delle precedenti**
- 

**9. Quale output si ottiene invocando il metodo g?**

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX**
  - B. XZ**
  - C. ZZ**
  - D. Errore a tempo di compilazione**
  - E. Errore a tempo di esecuzione**
- 

**10. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.**
  - B. Una classe interna non può essere dichiarata final.**
  - C. Una classe interna non può essere dichiarata static.**
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.**
  - E. Nessuna delle precedenti**
- 

**11. Date le dichiarazioni:**

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

## 12. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

## 13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
    }
}
```

```
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

## 14. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

## 15. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]){
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.

- B.** Viene lanciata una  
NullPointerException.  
**C.** Viene lanciata una  
ArrayIndexOutOfBoundsException.  
**D. null**  
**E. Nessuna delle precedenti.**
- 

**16. Date le dichiarazioni:**

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

---

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono  
corretti a tempo di compilazione.

- A.** r = n;  
**B.** n = r;  
**C.** z = r;  
**D.** r = z;  
**E.** z = n;



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 6**

1. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
    }
}
```

```
catch( Exception i ) {
    System.out.print(2);
}
catch( MyExc3 a ) {
}
finally {
    System.out.print(3);
    throw( new MyExc2() );
}
static void p()
throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}
```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
    }
}
```

```

}
catch( MyExc2 g ) {
    System.out.print(2);
}
catch( MyExc3 w ) {
}
}
static void m() {
try {
    System.out.print(3);
    throw( new MyExc3() );
}
catch( MyExc2 f ) {
    System.out.print(4);
}
catch( MyExc3 v ) {
    System.out.print(5);
}
catch( Exception i ) {
}
finally {
    System.out.print(6);
    throw( new MyExc1() );
}
}
}

```

- A. 135  
B. 1356Exception in thread "main" MyExc1  
C. Errore a tempo di compilazione  
D. 1356  
E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {

```

```

    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

5. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;  
B. q = (Error) m;  
C. m = (Boolean) q;  
D. m = (Boolean) u;  
E. Nessuno dei precedenti
- 

6. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;  
B. abstract int n = 0;  
C. protected int n = 0;  
D. public int n;  
E. Nessuna delle precedenti
- 

7. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

9. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
    }  
}
```

```
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

10. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

11. Date le dichiarazioni:

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {

```

```

            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**14. Quale output si ottiene invocando il metodo g?**

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

**15. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.

- C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

16. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";
```

```
double massIdx = 1.98910;  
  
public static void main(String[] args){  
    Star s1 = new Star();  
    Sun s2 = new Sun();  
    s1 = s2;  
    s2 = null; /*QUI*/  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 7**

1. Dato il seguente codice:

```
1. public interface I {
2.     int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

2. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

3. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}
```

```

}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
    }
}

```

```

        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3
B. Fiat Opel Corsa Citroen C3
C. Citroen Citroen Citroen C3
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

8. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

10. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

Super n;

B1 r;

C2 z;

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

11. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
    }
}

```

```

        catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}
}

```

- A. 135  
 B. 1356Exception in thread "main" MyExc1  
 C. Errore a tempo di compilazione  
 D. 1356  
 E. Nessuna delle precedenti
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145

- B. 145632  
 C. Errore a tempo di compilazione  
 D. 145623Exception in thread "main" MyExc2  
 E. Nessuna delle precedenti
- 

**14. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
 B. Quando due metodi hanno lo stesso nome si ha overloading  
 C. Non tutti i tipi di eccezioni estendono la classe Object  
 D. Un array non possiede dei membri  
 E. Due metodi possono differire per il tipo di ritorno
- 

**15. Quanti oggetti *al massimo possono essere deallocated al punto indicato?***

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args){
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

A. 1
B. 2
C. 3
D. 4
E. Nessuna delle precedenti

```

---

**16. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
}

```

```

public String id() {
    return producer + " ";
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {

```

```

        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- 
- A. Fiat Opel Citroen C3**  
**B. Fiat Opel Corsa Citroen C3**  
**C. Citroen Citroen Citroen C3**  
**D. Errore a tempo di compilazione.**  
**E. Errore a tempo di esecuzione.**



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 8**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

2. Quale output si ottiene invocando il metodo g?

```
class X {
```

```
void f(X x) {
    System.out.println("X");
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
```

```

try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}
}

```

- A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
- 

4. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)  
B. public int f( int j)  
C. private void f( int j)  
D. public static void f( int j)  
E. Nessuna delle precedenti
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
B. Quando due metodi hanno lo stesso nome si ha overloading  
C. Non tutti i tipi di eccezioni estendono la classe Object  
D. Un array non possiede dei membri  
E. Due metodi possono differire per il tipo di ritorno
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

7. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;  
B. abstract int n = 0;  
C. protected int n = 0;  
D. public int n;  
E. Nessuna delle precedenti
- 

8. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
B. Viene lanciata una NullPointerException.  
C. Viene lanciata una ArrayIndexOutOfBoundsException.  
D. null  
E. Nessuna delle precedenti.
- 

9. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;  
B. q = (Error) m;  
C. m = (Boolean) q;  
D. m = (Boolean) u;  
E. Nessuno dei precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

```

```

    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

11. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011  
B. 111  
C. 110  
D. 010  
E. 100
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**13. Date le dichiarazioni:**

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**15. Quanti oggetti *al massimo possono essere deallocati al punto indicato?***

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {

```

```
String name = "Sun";
double massIdx = 1.98910;

public static void main(String[] args) {
    Star s1 = new Star();
    Sun s2 = new Sun();
    s1 = s2;
    s2 = null;      /*QUI*/
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

**16. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 9**

1. Data la seguente classe:

```
1. public class C {
2.     public void f( int i) {}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011
- B. 111

- C. 110
- D. 010
- E. 100

3. Quanti oggetti **al massimo possono essere deallocati al punto indicato?**

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

public static void main(String[] args) {
    Star s1 = new Star();
    Sun s2 = new Sun();
    s1 = s2;
    s2 = null; /*QUI*/
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}
```

```

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
}

public static String id() {
    return producer + " " + model + " ";
}

}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

#### 5. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}

class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}

```

- A. XX**
  - B. XZ**
  - C. ZZ**
  - D. Errore a tempo di compilazione**
  - E. Errore a tempo di esecuzione**
- 

#### 6. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

#### e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;**
  - B. n = r;**
  - C. z = r;**
  - D. r = z;**
  - E. z = n;**
- 

#### 7. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri**
  - B. Quando due metodi hanno lo stesso nome si ha overloading**
  - C. Non tutti i tipi di eccezioni estendono la classe Object**
  - D. Un array non possiede dei membri**
  - E. Due metodi possono differire per il tipo di ritorno**
- 

#### 8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
    }
}

```

```

        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

### 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2**
- B. 157**
- C. 157342**
- D. Errore a tempo di compilazione**
- E. Nessuna delle precedenti**

### 10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

### 11. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;**
- B. q = (Error) m;**
- C. m = (Boolean) q;**

- D. `m = (Boolean) u;`  
E. Nessuno dei precedenti

---

**12. Dato il seguente codice:**

```
1. public interface I {  
2.     int n = 0;  
3. }
```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. `final int n = 0;`  
B. `abstract int n = 0;`  
C. `protected int n = 0;`  
D. `public int n;`  
E. Nessuna delle precedenti

---

**13. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 f ) {  
            System.out.print(4);  
        }  
        catch( MyExc3 v ) {  
            System.out.print(5);  
        }  
        catch( Exception i ) {  
        }  
    finally {  
        System.out.print(6);  
        throw( new MyExc1() );  
    }  
}
```

**A. 135**

- B. 1356Exception in thread "main" MyExc1  
C. Errore a tempo di compilazione  
D. 1356  
E. Nessuna delle precedenti

---

**14. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception i ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 a ) {  
        }  
    finally {  
        System.out.print(3);  
        throw( new MyExc2() );  
    }  
}  
static void p()  
    throws Exception {  
    try {  
        System.out.print(4);  
        throw( new MyExc1() );  
    }  
    catch( MyExc2 f ) {  
    }  
    catch( MyExc3 b ) {  
    }  
    catch( MyExc1 f ) {  
        System.out.print(5);  
    }  
    finally {  
        System.out.print(6);  
        throw( new MyExc3() );  
    }  
}
```

- A. 145**  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti

---

**15. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.**

- B. Una classe interna non può essere dichiarata final.**
  - C. Una classe interna non può essere dichiarata static.**
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.**
  - E. Nessuna delle precedenti**
- 

**16. Qual'è l'output di questo programma?**

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];
```

- ```
        System.out.println(c[6]);  
    }  
}
```
- A. Errore a tempo di compilazione per errata dichiarazione di un array.**
  - B. Viene lanciata una NullPointerException.**
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.**
  - D. null**
  - E. Nessuna delle precedenti.**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 10**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

2. Data la seguente classe:

1. public class C {

2. public void f( int i){}
3.
4. }

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

4. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 011  
B. 111  
C. 110  
D. 010  
E. 100
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145  
B. 145632

- C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.  
B. Una classe interna non può essere dichiarata final.  
C. Una classe interna non può essere dichiarata static.  
D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.  
E. Nessuna delle precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.

- 
8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car() {}  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

9. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
B. Viene lanciata una NullPointerException.

- C. Viene lanciata una ArrayIndexOutOfBoundsException.  
D. null  
E. Nessuna delle precedenti.
- 

10. Date le dichiarazioni:

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;  
B. n = r;  
C. z = r;  
D. r = z;  
E. z = n;
- 

11. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;  
B. q = (Error) m;  
C. m = (Boolean) q;  
D. m = (Boolean) u;  
E. Nessuno dei precedenti
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {
```

```

        return producer + " ";
    }

}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

13. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

14. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

16. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
public static void main(String[] args) {
```

```
    Star s1 = new Star();  
    Sun s2 = new Sun();  
    s1 = s2;  
    s2 = null; /*QUI*/  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 11

1. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
}  
void g() {  
    Y y = new Y();  
    Object o = new Z();  
    f(y); f(o);  
}
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

2. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception i ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 a ) {  
        }  
        finally {  
            System.out.print(3);  
            throw( new MyExc2() );  
        }  
    }  
    static void p()  
        throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 f ) {  
        }  
        catch( MyExc3 b ) {  
        }  
        catch( MyExc1 f ) {  
            System.out.print(5);  
        }  
        finally {  
            System.out.print(6);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

4. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {

```

```

            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342

- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

9. Dato il seguente codice:

```
1. public interface I {
2.     int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

---

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3

- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

- 
11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car() {}  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

- 
12. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.

- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

- 
13. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args){  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }  
}  
  
A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
```

- 
14. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

- 
15. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading

- C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

**16. Date le dichiarazioni:**

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

---

Super n;  
B1 r;  
C2 z;

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 12**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

2. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata **private**.
  - B. Una classe interna non può essere dichiarata **final**.
  - C. Una classe interna non può essere dichiarata **static**.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

4. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
```

```

        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

5. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

6. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri

- B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

8. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

9. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145**  
**B. 145632**  
**C. Errore a tempo di compilazione**  
**D. 145623Exception in thread "main" MyExc2**  
**E. Nessuna delle precedenti**
- 

## 11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {

```

```

            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

## 12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);

```

```

        if (true) throw( new MyExc3() );
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 1574Exception in thread "main" MyExc2**  
**B. 157**  
**C. 157342**  
**D. Errore a tempo di compilazione**  
**E. Nessuna delle precedenti**
- 

### 13. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX**  
**B. XZ**  
**C. ZZ**  
**D. Errore a tempo di compilazione**  
**E. Errore a tempo di esecuzione**
- 

### 14. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;**

- B. q = (Error) m;**  
**C. m = (Boolean) q;**  
**D. m = (Boolean) u;**  
**E. Nessuno dei precedenti**
- 

### 15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3
B. Fiat Opel Corsa Citroen C3
C. Citroen Citroen Citroen C3
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

### 16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
}

```

```

public String id() {
    return producer + " ";
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
    }
}

```

```

Vehicle c = new Car("Opel","Corsa");
Car d = new Car("Citroen","C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 13**

1. Qual è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una `NullPointerException`.
- C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
- D. `null`
- E. Nessuna delle precedenti.

2. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. `final int n = 0;`
- B. `abstract int n = 0;`
- C. `protected int n = 0;`
- D. `public int n;`
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
}
```

```
public static String id() {  
    return producer + " ";  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}  
  
    public static String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata `private`.
- B. Una classe interna non può essere dichiarata `final`.
- C. Una classe interna non può essere dichiarata `static`.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

5. Quale output si ottiene invocando il metodo `q`?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {

```

```

            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

#### 8. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
}

```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}   
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

10. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args){  
        Star s1 = new Star();
```

```
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 f ) {  
            System.out.print(4);  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(5);  
        }  
    }  
}
```

```

        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

#### 13. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;**  
**B. q = (Error) m;**  
**C. m = (Boolean) q;**  
**D. m = (Boolean) u;**  
**E. Nessuno dei precedenti**
- 

#### 14. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)**  
**B. public int f( int j)**  
**C. private void f( int j)**  
**D. public static void f( int j)**  
**E. Nessuna delle precedenti**
- 

#### 15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3
B. Fiat Opel Corsa Citroen C3
C. Citroen Citroen Citroen C3
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

#### 16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {

```

```
        System.out.print(5);
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
static void p()
throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
```

---

A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 14**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
    try {
        System.out.print(5);
        if (true) throw( new MyExc3() );
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

3. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args){
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

## 5. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX**

**B. X Z**

**C. Z Z**

**D. Errore a tempo di compilazione**

**E. Errore a tempo di esecuzione**

---

## 6. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;**
  - B. abstract int n = 0;**
  - C. protected int n = 0;**
  - D. public int n;**
  - E. Nessuna delle precedenti**
- 

## 7. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

**e le dichiarazioni di variabile:**

```

Super n;
B1 r;
C2 z;

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;**
  - B. n = r;**
  - C. z = r;**
  - D. r = z;**
  - E. z = n;**
- 

## 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
    }
}

```

```

}
catch( MyExc3 a ) {
}
finally {
    System.out.print(3);
    throw( new MyExc2() );
}
}

static void p()
throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}
}

```

- A. 145**  
**B. 145632**  
**C. Errore a tempo di compilazione**  
**D. 145623Exception in thread "main" MyExc2**  
**E. Nessuna delle precedenti**
- 

## 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
    }
}

```

```

}
catch( MyExc3 v ) {
    System.out.print(5);
}
catch( Exception i ) {
}
finally {
    System.out.print(6);
    throw( new MyExc1() );
}
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

## 10. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri**  
**B. Quando due metodi hanno lo stesso nome si ha overloading**  
**C. Non tutti i tipi di eccezioni estendono la classe Object**  
**D. Un array non possiede dei membri**  
**E. Due metodi possono differire per il tipo di ritorno**
- 

## 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public static String id() {
        return producer + " " ;
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
}

```

```

Vehicle v = new Vehicle("Fiat","500");           }
Vehicle c = new Car("Opel","Corsa");            }
Car d = new Car("Citroen","C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

## 12. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.**
  - B. Viene lanciata una NullPointerException.**
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.**
  - D. null**
  - E. Nessuna delle precedenti.**
- 

## 13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

## 14. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011**
  - B. 111**
  - C. 110**
  - D. 010**
  - E. 100**
- 

## 15. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 15**

1. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
```

```

        System.out.print(1);
        m();
    }
    catch( MyExc2 g ) {
        System.out.print(2);
    }
    catch( MyExc3 w ) {
    }
}
static void m() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    catch( MyExc2 f ) {
        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

##### 5. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

##### e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;**  
**B. n = r;**  
**C. z = r;**  
**D. r = z;**  
**E. z = n;**
- 

##### 6. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;**  
**B. q = (Error) m;**  
**C. m = (Boolean) q;**  
**D. m = (Boolean) u;**  
**E. Nessuno dei precedenti**
- 

##### 7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " " ;
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3
B. Fiat Opel Corsa Citroen C3
C. Citroen Citroen Citroen C3
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

**8. Quale output si ottiene invocando il metodo g?**

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
}  
void g() {  
    Y y = new Y();  
    Object o = new Z();  
    f(y); f(o);  
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

**9. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(){}  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
    }  
}
```

```
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**10. Dato il seguente codice:**

```
1. public interface I {  
2.     int n = 0;  
3. }
```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

**11. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
            finally {  
                System.out.print(3);  
                throw( new MyExc2() );  
            }  
        }  
        static void p()  
            throws Exception {  
            try {  
                System.out.print(4);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 f ) {  
            }  
            catch( MyExc3 b ) {  
            }  
        }  
}
```

```

        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145  
 B. 145632  
 C. Errore a tempo di compilazione  
 D. 145623Exception in thread "main" MyExc2  
 E. Nessuna delle precedenti
- 

## 12. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
 B. Quando due metodi hanno lo stesso nome si ha overloading  
 C. Non tutti i tipi di eccezioni estendono la classe Object  
 D. Un array non possiede dei membri  
 E. Due metodi possono differire per il tipo di ritorno
- 

## 13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);

```

```

            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
 B. 157  
 C. 157342  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

## 14. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)  
 B. public int f( int j)  
 C. private void f( int j)  
 D. public static void f( int j)  
 E. Nessuna delle precedenti
- 

## 15. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
 B. Viene lanciata una NullPointerException.  
 C. Viene lanciata una ArrayIndexOutOfBoundsException.  
 D. null  
 E. Nessuna delle precedenti.
- 

## 16. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {}  
    if(i1 == i2) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(f2 == f1) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(i3 == i4) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 16**

1. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}
    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
```

```
Vehicle c = new Car("Opel","Corsa");
Car d = new Car("Citroen","C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

4. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;
}

public static void main(String[] args){}
```

```

Star s1 = new Star();
Sun s2 = new Sun();
s1 = s2;
s2 = null; /*QUI*/
}
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

#### 5. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

#### 6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {

```

```

        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

**B. 145632**

**C. Errore a tempo di compilazione**

**D. 145623Exception in thread "main" MyExc2**

**E. Nessuna delle precedenti**

**8. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

**A. 1574Exception in thread "main" MyExc2**

**B. 157**

**C. 157342**

**D. Errore a tempo di compilazione**

**E. Nessuna delle precedenti**

**9. Quale output si ottiene invocando il metodo q?**

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
}

```

```

private Integer i4 = new Integer(20);
void q() {
    m(i1, f1, new Integer(20));
}
void m(Integer i2, Object f2, Integer i3) {
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

**A. 011**

**B. 111**

**C. 110**

**D. 010**

**E. 100**

**10. Dire quale delle seguenti affermazioni è vera:**

**A. Una classe interna non può essere dichiarata private.**

**B. Una classe interna non può essere dichiarata final.**

**C. Una classe interna non può essere dichiarata static.**

**D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.**

**E. Nessuna delle precedenti**

**11. Dato il seguente codice:**

1. public interface I {
2. int n = 0;
3. }

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

**A. final int n = 0;**

**B. abstract int n = 0;**

**C. protected int n = 0;**

**D. public int n;**

**E. Nessuna delle precedenti**

## 12. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

## 13. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

## 14. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri

## E. Due metodi possono differire per il tipo di ritorno

## 15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
            System.out.print(3);
        }
    }
    static void m() {
        try {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(5);
        }
        catch( MyExc3 v ) {
            System.out.print(6);
        }
        catch( Exception i ) {
            finally {
                System.out.print(7);
                throw( new MyExc1() );
            }
        }
    }
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356
- E. Nessuna delle precedenti

## 16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public static String id() {
```

```

        return producer + " ";
    }

}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
    }
}

Car d = new Car("Citroen","C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}

```

A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 17**

1. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

2. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

3. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
            finally {  
                System.out.print(3);  
                throw( new MyExc2() );  
            }  
        }  
        static void p()  
        throws Exception {  
            try {  
                System.out.print(4);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 f ) {  
            }  
            catch( MyExc3 b ) {  
            }  
            catch( MyExc1 f ) {  
                System.out.print(5);  
            }  
            finally {  
                System.out.print(6);  
                throw( new MyExc3() );  
            }  
        }  
    }
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
```

```
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Dire quale delle seguenti affermazioni è vera:

- 
- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

9. Quanti oggetti **al massimo possono essere deallocati al punto indicato?**

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args){  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

10. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 f ) {  
            System.out.print(4);  
        }  
        catch( MyExc3 v ) {  
            System.out.print(5);  
        }  
        catch( Exception i ) {  
        }  
        finally {  
            System.out.print(6);  
            throw( new MyExc1() );  
        }  
    }  
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

12. Date le dichiarazioni:

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A.  $r = n;$
- B.  $n = r;$
- C.  $z = r;$
- D.  $r = z;$
- E.  $z = n;$

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
    }
}
```

```
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

**15. Qual'è l'output di questo programma?**

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

**16. Quale output si ottiene invocando il metodo q?**

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {}  
    if(i1 == i2) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(f2 == f1) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(i3 == i4) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 18**

1. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
    }
}
```

```
}  
catch( MyExc3 a ) {  
}  
finally {  
    System.out.print(3);  
    throw( new MyExc2() );  
}  
}  
static void p()  
throws Exception {  
try {  
    System.out.print(4);  
    throw( new MyExc1() );  
}  
catch( MyExc2 f ) {  
}  
catch( MyExc3 b ) {  
}  
catch( MyExc1 f ) {  
    System.out.print(5);  
}  
finally {  
    System.out.print(6);  
    throw( new MyExc3() );  
}  
}  
}  
A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
```

---

4. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

---

5. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}  
  
    public static String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

7. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
        }  
        catch( MyExc1 v ) {  
            System.out.print(2);  
        }  
        catch( Exception d ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc2() );  
        }  
    }  
    static void q()  
        throws Exception {  
        try {  
            System.out.print(5);  
            if (true) throw( new MyExc3() );  
            else throw( new MyExc1() );  
        }  
        catch( MyExc1 z ) {  
            System.out.print(6);  
        }  
    }  
}
```

```

        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
 B. 157  
 C. 157342  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

9. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.  
 B. Una classe interna non può essere dichiarata final.  
 C. Una classe interna non può essere dichiarata static.  
 D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.  
 E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
}

```

- A. X X  
 B. X Z  
 C. Z Z  
 D. Errore a tempo di compilazione  
 E. Errore a tempo di esecuzione
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

13. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

---

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

15. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 011
- B. 111
- C. 110
- D. 010
- E. 100

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
        }  
    }  
}
```

```
}

static void m() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    catch( MyExc2 f ) {
        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}
```

---

- A. 135**
- B. 1356Exception in thread "main" MyExc1**
- C. Errore a tempo di compilazione**
- D. 1356**
- E. Nessuna delle precedenti**



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 19**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public static String id() {
        return producer + " ";
    }
}
class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public static String id() {
        return producer + " " + model + " ";
    }
}
class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object

- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

---

4. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

---

5. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
    void g() {  
        Y y = new Y();  
        Object o = new Z();  
        f(y); f(o);  
    }  
}
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

6. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.

- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(){}  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

8. Date le dichiarazioni:

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}   
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}
```

```
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 f ) {  
            System.out.print(4);  
        }  
        catch( MyExc3 v ) {  
            System.out.print(5);  
        }  
        catch( Exception i ) {  
            System.out.print(6);  
            throw( new MyExc1() );  
        }  
    }  
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356
- E. Nessuna delle precedenti

---

11. Qual è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

---

12. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

13. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1**
- B. 2**
- C. 3**
- D. 4**

#### E. Nessuna delle precedenti

---

14. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;**
  - B. q = (Error) m;**
  - C. m = (Boolean) q;**
  - D. m = (Boolean) u;**
  - E. Nessuno dei precedenti**
- 

15. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)**
  - B. public int f( int j)**
  - C. private void f( int j)**
  - D. public static void f( int j)**
  - E. Nessuna delle precedenti**
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

```
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

static void p()
throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc2 f ) {
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc1 f ) {
        System.out.print(5);
    }
}
```

---

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 20**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

2. Quale output si ottiene invocando il metodo g?

```
class X {
```

```
void f(X x) {
    System.out.println("X");
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

4. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

5. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
        }  
        catch( MyExc1 v ) {  
            System.out.print(2);  
        }  
        catch( Exception d ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc2() );  
        }  
    }  
    static void q()  
        throws Exception {  
        try {  
            System.out.print(5);  
            if (true) throw( new MyExc3() );  
            else throw( new MyExc1() );  
        }
```

```
        catch( MyExc1 z ) {  
            System.out.print(6);  
        }  
        finally {  
            System.out.print(7);  
        }  
    }  
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

8. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args){  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

9. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

---

**10. Date le dichiarazioni:**

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

---

**11. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
  
    class Car extends Vehicle {  
        Car(String p, String m) {super(p,m);}  
  
        public static String id() {  
            return producer + " " + model + " ";  
        }  
    }  
}
```

```
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat", "500");  
        Vehicle c = new Car("Opel", "Corsa");  
        Car d = new Car("Citroen", "C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

**12. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(){}  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat", "500");  
        Vehicle c = new Car("Opel", "Corsa");  
        Car d = new Car("Citroen", "C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

**13. Quale output si ottiene invocando il metodo q?**

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 011
- B. 111
- C. 110
- D. 010
- E. 100

---

**14. Qual è l'output di questo programma?**

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

---

**15. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " " ;  
    }  
  
    class Car extends Vehicle {  
        Car(String p, String m) {super(p,m);}  
  
        public String id() {  
            return producer + " " + model + " ";  
        }  
    }  
  
    class Main {  
        public static void main(String[] args) {  
            Vehicle v = new Vehicle("Fiat","500");  
            Vehicle c = new Car("Opel","Corsa");  
            Car d = new Car("Citroen","C3");  
            System.out.print(v.id());  
            System.out.print(c.id());  
            System.out.print(d.id());  
        }  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

**16. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception i ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 a ) {  
        }  
        finally {  
            System.out.print(3);  
            throw( new MyExc2() );  
        }  
    }  
}
```

```
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

static void p()
throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc2 f ) {
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc1 f ) {
        System.out.print(5);
    }
}
```

---

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 21**

1. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

2. Dato il seguente codice:

```
1. public interface I {
2.     int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

3. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
```

```
Y y = new Y();
Object o = new Z();
f(y); f(o);
}
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}
```

```
    }
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

5. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {

```

```
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {

```

```

}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}

```

- A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
- 

8. Qual è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
B. Viene lanciata una NullPointerException.  
C. Viene lanciata una ArrayIndexOutOfBoundsException.  
D. null  
E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
}

```

```

Car(String p, String m) {super(p,m);}

public String id() {
    return producer + " " + model + " ";
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

10. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;  
B. n = r;  
C. z = r;  
D. r = z;  
E. z = n;
- 

11. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;
}

```

```

public static void main(String[] args) {
    Star s1 = new Star();
    Sun s2 = new Sun();
    s1 = s2;
    s2 = null; /*QUI*/
}

```

- A. 1
- B. 2
- C. 3
- D. 4

#### E. Nessuna delle precedenti

---

#### 12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione

#### E. Nessuna delle precedenti

---

#### 13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 14. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

#### 15. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 22**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Dato il seguente codice:

```
1. public interface I {
2.     int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

3. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

#### 5. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();

```

```

        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

#### 6. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
B. Quando due metodi hanno lo stesso nome si ha overloading  
C. Non tutti i tipi di eccezioni estendono la classe Object  
D. Un array non possiede dei membri  
E. Due metodi possono differire per il tipo di ritorno
- 

#### 7. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

#### e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;  
B. n = r;  
C. z = r;  
D. r = z;  
E. z = n;
- 

#### 8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

## 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {

```

```

            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135**
  - B. 1356Exception in thread "main" MyExc1**
  - C. Errore a tempo di compilazione**
  - D. 1356**
  - E. Nessuna delle precedenti**
- 

## 10. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.**
  - B. Viene lanciata una NullPointerException.**
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.**
  - D. null**
  - E. Nessuna delle precedenti.**
- 

## 11. Data la seguente classe:

1. public class C {
2. public void f( int i){}
3. }
4. }

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)**
- B. public int f( int j)**
- C. private void f( int j)**
- D. public static void f( int j)**

**E. Nessuna delle precedenti**

---

**12. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

```

```

    }

}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

**14. Date le dichiarazioni:**

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;**
  - B. q = (Error) m;**
  - C. m = (Boolean) q;**
  - D. m = (Boolean) u;**
  - E. Nessuno dei precedenti**
- 

**15. Quale output si ottiene invocando il metodo q?**

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        }
    }
}

```

```
    } else {
        System.out.print(0);
    }
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 011
- B. 111
- C. 110

D. 010

E. 100

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 23**

1. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

2. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception { }
```

```

try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
finally {
    System.out.print(7);
}
}

```

- A. 1574Exception in thread "main" MyExc2  
 B. 157  
 C. 157342  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
 B. Fiat Opel Corsa Citroen C3

- C. Citroen Citroen Citroen C3  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

6. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;  
 B. abstract int n = 0;  
 C. protected int n = 0;  
 D. public int n;  
 E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
        }
    }
}

```

```

        throw( new MyExc3() );
    }
}

```

- A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
- 

8. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)  
B. public int f( int j)  
C. private void f( int j)  
D. public static void f( int j)  
E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
    }
}

```

```

        catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 135  
B. 1356Exception in thread "main" MyExc1  
C. Errore a tempo di compilazione  
D. 1356  
E. Nessuna delle precedenti
- 

10. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args){
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

```

```

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

## 12. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

## 13. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}

```

```

void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

## 14. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

## 15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " " + model + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
    }
}

```

```
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Date le dichiarazioni:

Boolean m;

---

```
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 24**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

- 
2. Data la seguente classe:

1. public class C {
2. public void f( int i){}
3. }
4. }

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

- 
3. Dato il seguente codice:

1. public interface I {
2. int n = 0;
3. }

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

- 
4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
        }
    }
}
```

```

        throw( new MyExc2() );
    }
}

static void q()
    throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
finally {
    System.out.print(7);
}
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 5. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.  
B. Una classe interna non può essere dichiarata final.  
C. Una classe interna non può essere dichiarata static.  
D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.  
E. Nessuna delle precedenti
- 

#### 6. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

#### 7. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

#### e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;  
B. n = r;  
C. z = r;  
D. r = z;  
E. z = n;
- 

#### 8. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
B. Quando due metodi hanno lo stesso nome si ha overloading  
C. Non tutti i tipi di eccezioni estendono la classe Object  
D. Un array non possiede dei membri  
E. Due metodi possono differire per il tipo di ritorno
- 

#### 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {

```

```

        System.out.print(2);
    }
    catch( MyExc3 a ) {
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
static void p()
throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}

```

- A. 145**  
**B. 145632**  
**C. Errore a tempo di compilazione**  
**D. 145623Exception in thread "main" MyExc2**  
**E. Nessuna delle precedenti**
- 

## 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {

```

```

            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

## 11. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;**  
**B. q = (Error) m;**  
**C. m = (Boolean) q;**  
**D. m = (Boolean) u;**  
**E. Nessuno dei precedenti**
- 

## 12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " ";
    }
}
class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
}

```

```

public String id() {
    return producer + " " + model + " ";
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

### 13. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX**
  - B. XZ**
  - C. ZZ**
  - D. Errore a tempo di compilazione**
  - E. Errore a tempo di esecuzione**
- 

### 14. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.**
  - B. Viene lanciata una NullPointerException.**
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.**
  - D. null**
  - E. Nessuna delle precedenti.**
- 

### 15. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011**
  - B. 111**
  - C. 110**
  - D. 010**
  - E. 100**
- 

### 16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

```

```

}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}
    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
}

```

```

        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 25**

1. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

2. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

3. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " " ;
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args) {  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

6. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

7. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

8. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
  
    class Car extends Vehicle {  
        Car(String p, String m) {super(p,m);}  
  
        public static String id() {  
            return producer + " " + model + " ";  
        }  
    }  
  
    class Main {  
        public static void main(String[] args) {  
            Vehicle v = new Vehicle("Fiat","500");  
            Vehicle c = new Car("Opel","Corsa");  
            Car d = new Car("Citroen","C3");  
            System.out.print(v.id());  
            System.out.print(c.id());  
            System.out.print(d.id());  
        }  
    }  
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

#### 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

```

```

Vehicle(String p, String m) {
    producer = p;
    model = m;
}

public String id() {
    return producer + " ";
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 12. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        }
    }
}

```

```

    } else {
        System.out.print(0);
    }
}
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

### 13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
                p();
            }
            catch( Exception i ) {
                System.out.print(2);
            }
            catch( MyExc3 a ) {
            }
            finally {
                System.out.print(3);
                throw( new MyExc2() );
            }
        }
        static void p()
            throws Exception {
            try {
                System.out.print(4);
                throw( new MyExc1() );
            }
            catch( MyExc2 f ) {
            }
            catch( MyExc3 b ) {
            }
            catch( MyExc1 f ) {
                System.out.print(5);
            }
            finally {
                System.out.print(6);
                throw( new MyExc3() );
            }
        }
}

```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione

D. 145623Exception in thread "main" MyExc2

E. Nessuna delle precedenti

---

### 14. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

### 15. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

### 16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
                q();
            }
            catch( MyExc1 v ) {
                System.out.print(2);
            }
            catch( Exception d ) {

```

```
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc2() );
    }
}

static void q()
    throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
        }
    finally {
        System.out.print(7);
    }
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 26**

1. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

2. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
}
```

```
public String id() {
    return producer + " ";
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Qual è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

- 
5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}   
  
    public static String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

6. Data la seguente classe:

1. public class C {
2. public void f( int i){}
3. }
4. }

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)

- 
- E. Nessuna delle precedenti

- 
7. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
}  
void g() {  
    Y y = new Y();  
    Object o = new Z();  
    f(y); f(o);  
}
```

- A. XX  
B. XZ  
C. ZZ  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
            finally {  
                System.out.print(3);  
                throw( new MyExc2() );  
            }  
        }  
        static void p()  
        throws Exception {  
            try {  
                System.out.print(4);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 f ) {
```

```

}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}
}

```

- A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
- 

**9. Quale output si ottiene invocando il metodo q?**

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011  
B. 111  
C. 110  
D. 010  
E. 100
- 

**10. Dato il seguente codice:**

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;  
B. abstract int n = 0;  
C. protected int n = 0;  
D. public int n;  
E. Nessuna delle precedenti
- 

**11. Quanti oggetti *al massimo possono essere deallocati al punto indicato?***

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args){
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

A. 1
B. 2
C. 3
D. 4
E. Nessuna delle precedenti

```

---

**12. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
    }
}

```

```

        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
    try {
        System.out.print(5);
        if (true) throw( new MyExc3() );
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 1574Exception in thread "main" MyExc2**  
**B. 157**  
**C. 157342**  
**D. Errore a tempo di compilazione**  
**E. Nessuna delle precedenti**
- 

### 13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
    }
}

```

```

        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

### 14. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;**  
**B. q = (Error) m;**  
**C. m = (Boolean) q;**  
**D. m = (Boolean) u;**  
**E. Nessuno dei precedenti**
- 

### 15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {

```

```
Vehicle v = new Vehicle("Fiat","500");
Vehicle c = new Car("Opel","Corsa");
Car d = new Car("Citroen","C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 27

1. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

2. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}
```

```
class Car extends Vehicle {  
    Car(){ }  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

5. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
            System.out.print(3);
        }
    }
    static void m() {
        try {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(5);
        }
        catch( MyExc3 v ) {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 011  
 B. 111  
 C. 110  
 D. 010  
 E. 100
- 

12. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;  
 B. abstract int n = 0;  
 C. protected int n = 0;  
 D. public int n;  
 E. Nessuna delle precedenti
- 

13. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1

- B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.  
 B. Una classe interna non può essere dichiarata final.  
 C. Una classe interna non può essere dichiarata static.  
 D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.  
 E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
 B. Fiat Opel Corsa Citroen C3  
 C. Citroen Citroen Citroen C3  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.

---

**16. Quale output si ottiene invocando il metodo g?**

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
}
```

```
Y y = new Y();  
Object o = new Z();  
f(y); f(o);  
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 28**

1. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

2. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
```

- A. XX

- B. X Z
  - C. Z Z
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
            System.out.print(3);
            throw( new MyExc3() );
        }
        static void m() {
            try {
                System.out.print(4);
                throw( new MyExc1() );
            }
            catch( MyExc2 f ) {
                System.out.print(5);
            }
            catch( MyExc3 v ) {
                System.out.print(6);
            }
            catch( Exception i ) {
            }
            finally {
                System.out.print(7);
                throw( new MyExc1() );
            }
        }
    }
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356

## E. Nessuna delle precedenti

### 4. Qual è l'output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 011
- B. 111
- C. 110
- D. 010
- E. 100

### 5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {
```

```
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat", "500");  
        Vehicle c = new Car("Opel", "Corsa");  
        Car d = new Car("Citroen", "C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Opel Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

### 6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
        }  
        catch( MyExc1 v ) {  
            System.out.print(2);  
        }  
        catch( Exception d ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc2() );  
        }  
    }  
    static void q()  
        throws Exception {  
        try {  
            System.out.print(5);  
            if (true) throw( new MyExc3() );  
            else throw( new MyExc1() );  
        }  
        catch( MyExc1 z ) {  
            System.out.print(6);  
        }  
        finally {  
            System.out.print(7);  
        }  
    }  
}
```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

**7. Dato il seguente codice:**

```
1. public interface I {  
2.     int n = 0;  
3. }
```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

---

**8. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car() {}  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

**9. Qual'è l'output di questo programma?**

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

---

**10. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception i ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 a ) {  
        }  
        finally {  
            System.out.print(3);  
            throw( new MyExc2() );  
        }  
    }  
    static void p()  
        throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 f ) {  
        }  
        catch( MyExc3 b ) {  
        }  
    }  
}
```

```

        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145  
 B. 145632  
 C. Errore a tempo di compilazione  
 D. 145623Exception in thread "main" MyExc2  
 E. Nessuna delle precedenti
- 

#### 11. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
 B. Quando due metodi hanno lo stesso nome si ha overloading  
 C. Non tutti i tipi di eccezioni estendono la classe Object  
 D. Un array non possiede dei membri  
 E. Due metodi possono differire per il tipo di ritorno
- 

#### 12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
    }
}

```

```

        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
 B. Fiat Opel Corsa Citroen C3  
 C. Citroen Citroen Citroen C3  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

#### 13. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;  
 B. n = r;  
 C. z = r;  
 D. r = z;  
 E. z = n;
- 

#### 14. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)  
 B. public int f( int j)  
 C. private void f( int j)  
 D. public static void f( int j)  
 E. Nessuna delle precedenti
- 

#### 15. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 29**

1. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
```

```
throws Exception {
    try {
        System.out.print(1);
        p();
    }
    catch( Exception i ) {
        System.out.print(2);
    }
    catch( MyExc3 a ) {
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
static void p()
throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc2 f ) {
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc1 f ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
        throw( new MyExc3() );
    }
}
```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

5. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabili:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;

E. z = n;

6. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011
- B. 111
- C. 110
- D. 010
- E. 100

7. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

8. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1**
  - B. 2**
  - C. 3**
  - D. 4**
  - E. Nessuna delle precedenti**
- 

## 9. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.**
  - B. Viene lanciata una NullPointerException.**
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.**
  - D. null**
  - E. Nessuna delle precedenti.**
- 

## 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {

```

```

            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2**
  - B. 157**
  - C. 157342**
  - D. Errore a tempo di compilazione**
  - E. Nessuna delle precedenti**
- 

## 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " " ;
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {

```

```

Vehicle v = new Vehicle("Fiat","500");
Vehicle c = new Car("Opel","Corsa");
Car d = new Car("Citroen","C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

**12. Dato il seguente codice:**

```

1. public interface I {
2.     int n = 0;
3. }

```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;**
  - B. abstract int n = 0;**
  - C. protected int n = 0;**
  - D. public int n;**
  - E. Nessuna delle precedenti**
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

**14. Date le dichiarazioni:**

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;**
  - B. q = (Error) m;**
  - C. m = (Boolean) q;**
  - D. m = (Boolean) u;**
  - E. Nessuno dei precedenti**
- 

**15. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
        }
    }
}

```

```
        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356

E. Nessuna delle precedenti

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 30**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

2. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
```

```
} }
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
```

```

public static void main(String[] args) {
    Vehicle v = new Vehicle("Fiat","500");
    Vehicle c = new Car("Opel","Corsa");
    Car d = new Car("Citroen","C3");
    System.out.print(v.id());
    System.out.print(c.id());
    System.out.print(d.id());
}
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
    }
}

```

```

    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}
}
A. 135
B. 1356Exception in thread "main" MyExc1
C. Errore a tempo di compilazione
D. 1356
E. Nessuna delle precedenti

```

---

6. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}

```

```

}
static void p()
    throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc2 f ) {
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc1 f ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
        throw( new MyExc3() );
    }
}
}

```

- A. 145  
 B. 145632  
 C. Errore a tempo di compilazione  
 D. 145623Exception in thread "main" MyExc2  
 E. Nessuna delle precedenti
- 

C. 110

D. 010

E. 100

---

**9. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

A. 1574Exception in thread "main" MyExc2

B. 157

C. 157342

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

**8. Quale output si ottiene invocando il metodo q?**

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011  
 B. 111

**10. Data la seguente classe:**

1. public class C {
2. public void f( int i){}
- 3.
4. }

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

12. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```
class Star {
    String name = "";
    double massIdx = 0;
}
```

```
class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

15. Dato il seguente codice:

```
1.  public interface I {
2.      int n = 0;
3.  }
```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

**16. Date le dichiarazioni:**

Boolean m;  
Error q;

---

```
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 31**

1. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
```

```
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

```

        }
    }

    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
    finally {
        System.out.print(6);
        throw( new MyExc3() );
    }
}

```

- A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
- 

5. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)  
B. public int f( int j)  
C. private void f( int j)  
D. public static void f( int j)  
E. Nessuna delle precedenti
- 

6. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;  
B. n = r;  
C. z = r;  
D. r = z;  
E. z = n;
- 

7. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
}

```

- A. XX  
B. XZ  
C. ZZ  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione
- 

8. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;  
B. abstract int n = 0;  
C. protected int n = 0;  
D. public int n;  
E. Nessuna delle precedenti
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 10. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

#### 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3
B. Fiat Opel Corsa Citroen C3
C. Citroen Citroen Citroen C3
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

#### 12. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
    }
}

```

```

        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

15. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
    }
}

```

```
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }

static void q()
    throws Exception {
    try {
        System.out.print(5);
        if (true) throw( new MyExc3() );
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}
```

A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 32

1. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

2. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata **private**.
- B. Una classe interna non può essere dichiarata **final**.
- C. Una classe interna non può essere dichiarata **static**.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;

- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        static void m() {  
            try {  
                System.out.print(4);  
                throw( new MyExc3() );  
            }  
            catch( MyExc2 f ) {  
                System.out.print(5);  
            }  
            catch( MyExc3 v ) {  
                System.out.print(6);  
                throw( new MyExc1() );  
            }  
        }  
    }  
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356
- E. Nessuna delle precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}   
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
        }  
}
```

```
        finally {  
            System.out.print(3);  
            throw( new MyExc2() );  
        }  
    }  
    static void p()  
        throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 f ) {  
        }  
        catch( MyExc3 b ) {  
        }  
        catch( MyExc1 f ) {  
            System.out.print(5);  
        }  
        finally {  
            System.out.print(6);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(){}  
    Car(String p, String m) {super(p,m);}   
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
    }  
}
```

```

Vehicle c = new Car("Opel", "Corsa");
Car d = new Car("Citroen", "C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

9. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

10. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

11. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        }
    }
}

```

```

    } else {
        System.out.print(0);
    }
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 011  
 B. 111  
 C. 110  
 D. 010  
 E. 100
- 

## 12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
 B. 157  
 C. 157342  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

## 13. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i )  
 B. public int f( int j )  
 C. private void f( int j )  
 D. public static void f( int j )  
 E. Nessuna delle precedenti
- 

## 14. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
 B. Quando due metodi hanno lo stesso nome si ha overloading  
 C. Non tutti i tipi di eccezioni estendono la classe Object  
 D. Un array non possiede dei membri  
 E. Due metodi possono differire per il tipo di ritorno
- 

## 15. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
 B. Viene lanciata una NullPointerException.  
 C. Viene lanciata una ArrayIndexOutOfBoundsException.  
 D. null  
 E. Nessuna delle precedenti.

- 
16. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
public static void main(String[] args) {
```

```
    Star s1 = new Star();  
    Sun s2 = new Sun();  
    s1 = s2;  
    s2 = null; /*QUI*/  
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 33**

1. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una `NullPointerException`.
  - C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
  - D. null
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
}
```

```
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
    finally {
        System.out.print(6);
        throw( new MyExc3() );
    }
}
```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " " ;
    }
}
class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}
    public String id() {
        return producer + " " + model + " ";
    }
}
class Main {
```

```

public static void main(String[] args) {
    Vehicle v = new Vehicle("Fiat", "500");
    Vehicle c = new Car("Opel", "Corsa");
    Car d = new Car("Citroen", "C3");
    System.out.print(v.id());
    System.out.print(c.id());
    System.out.print(d.id());
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
- 

E. Nessuna delle precedenti

---

#### 5. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

#### 6. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
    }
}

```

```

        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
    try {
        System.out.print(5);
        if (true) throw( new MyExc3() );
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

**8. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
    }
}

```

```

        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

**9. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?**

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

**10. Data la seguente classe:**

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)  
B. public int f( int j)  
C. private void f( int j)  
D. public static void f( int j)  
E. Nessuna delle precedenti
- 

**11. Date le dichiarazioni:**

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

**12. Dato il seguente codice:**

```

1. public interface I {
2.     int n = 0;
3. }

```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

**14. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

**15. Quale output si ottiene invocando il metodo q?**

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- B. 111**
- C. 110**
- D. 010**
- E. 100**

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;**
- B. n = r;**
- C. z = r;**
- D. r = z;**
- E. z = n;**

---

**16. Date le dichiarazioni:**

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

**e le dichiarazioni di variabile:** \_\_\_\_\_

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 34**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
            System.out.print(3);
        }
    }
    static void m() {
        try {
            System.out.print(4);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(5);
        }
        catch( MyExc3 v ) {
            System.out.print(6);
        }
        catch( Exception i ) {
            System.out.print(7);
        }
    }
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356
- E. Nessuna delle precedenti

3. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
```

```

void q() {
    m(i1, f1, new Integer(20));
}
void m(Integer i2, Object f2, Integer i3) {
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

4. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

5. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " " ;
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**7. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**8. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
```

```
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**9. Qual è l'output di questo programma?**

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

**10. Dato il seguente codice:**

1. public interface I {
2. int n = 0;
3. }

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

---

**11. Date le dichiarazioni:**

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

---

**12. Date le dichiarazioni:**

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

**e le dichiarazioni di variabile:**

```
Super n;
B1 r;
C2 z;
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

---

**13. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

---

**14. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

---

**15. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.**
  - B. Una classe interna non può essere dichiarata final.**
  - C. Una classe interna non può essere dichiarata static.**
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.**
  - E. Nessuna delle precedenti**
- 

**16. Data la seguente classe:**

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)**
  - B. public int f( int j)**
  - C. private void f( int j)**
  - D. public static void f( int j)**
  - E. Nessuna delle precedenti**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 35

1. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

2. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
    void g() {  
        Y y = new Y();  
        Object o = new Z();  
        f(y); f(o);  
    }  
}
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

3. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.

- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

4. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 011
- B. 111
- C. 110
- D. 010
- E. 100

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135  
B. 1356Exception in thread "main" MyExc1  
C. Errore a tempo di compilazione  
D. 1356  
E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
}

```

```

        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
- 

#### 7. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;  
B. q = (Error) m;  
C. m = (Boolean) q;  
D. m = (Boolean) u;  
E. Nessuno dei precedenti
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
}

```

```

try {
    System.out.print(1);
    q();
}
catch( MyExc1 v ) {
    System.out.print(2);
}
catch( Exception d ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
    throw( new MyExc2() );
}
}

static void q()
throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
finally {
    System.out.print(7);
}
}
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
B. Quando due metodi hanno lo stesso nome si ha overloading  
C. Non tutti i tipi di eccezioni estendono la classe Object  
D. Un array non possiede dei membri  
E. Due metodi possono differire per il tipo di ritorno
- 

10. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)  
B. public int f( int j)  
C. private void f( int j)  
D. public static void f( int j)  
E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3
B. Fiat Opel Corsa Citroen C3
C. Citroen Citroen Citroen C3
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

12. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**14. Date le dichiarazioni:**

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

**e le dichiarazioni di variabile:**

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

**15. Qual'è l'output di questo programma?**

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

**16. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- 
- A. Fiat Opel Citroen C3**  
**B. Fiat Opel Corsa Citroen C3**  
**C. Citroen Citroen Citroen C3**  
**D. Errore a tempo di compilazione.**  
**E. Errore a tempo di esecuzione.**



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 36

1. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

2. Dato il seguente codice:

```
1. public interface I {
2.     int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
```

```
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
}

public static String id() {
    return producer + " " + model + " ";
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

5. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

7. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

8. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

9. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

10. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

}
if(f2 == f1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
if(i3 == i4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 011**
  - B. 111**
  - C. 110**
  - D. 010**
  - E. 100**
- 

**11. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
- B. Fiat Opel Corsa Citroen C3**

**C. Citroen Citroen Citroen C3**

- D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

**12. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145**
  - B. 145632**
  - C. Errore a tempo di compilazione**
  - D. 145623Exception in thread "main" MyExc2**
  - E. Nessuna delle precedenti**
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
 B. 157  
 C. 157342  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {

```

```

            }
        }
        static void m() {
            try {
                System.out.print(3);
                throw( new MyExc3() );
            }
            catch( MyExc2 f ) {
                System.out.print(4);
            }
            catch( MyExc3 v ) {
                System.out.print(5);
            }
            catch( Exception i ) {
            }
            finally {
                System.out.print(6);
                throw( new MyExc1() );
            }
        }
    }
}

```

- A. 135  
 B. 1356Exception in thread "main" MyExc1  
 C. Errore a tempo di compilazione  
 D. 1356  
 E. Nessuna delle precedenti
- 

#### 15. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
 B. Viene lanciata una NullPointerException.  
 C. Viene lanciata una ArrayIndexOutOfBoundsException.  
 D. null  
 E. Nessuna delle precedenti.
- 

#### 16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
}

```

```

public String id() {
    return producer + " ";
}
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
}

public String id() {
    return producer + " " + model + " ";
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
    }
}

```

```

Vehicle c = new Car("Opel", "Corsa");
Car d = new Car("Citroen", "C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}

```

- 
- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 37**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri

- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

3. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
        }
    }
}
```

```

        throw( new MyExc3() );
    }
    catch( MyExc2 f ) {
        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 135  
B. 1356Exception in thread "main" MyExc1  
C. Errore a tempo di compilazione  
D. 1356  
E. Nessuna delle precedenti
- 

5. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX  
B. XZ  
C. ZZ  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)

```

```

        throws Exception {
    try {
        System.out.print(1);
        q();
    }
    catch( MyExc1 v ) {
        System.out.print(2);
    }
    catch( Exception d ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc2() );
    }
}
static void q()
    throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
finally {
    System.out.print(7);
}
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
    try {
        System.out.print(1);
        p();
    }
    catch( Exception i ) {
        System.out.print(2);
    }
    catch( MyExc3 a ) {
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}

```

```

        }
    }

    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145  
 B. 145632  
 C. Errore a tempo di compilazione  
 D. 145623Exception in thread "main" MyExc2  
 E. Nessuna delle precedenti
- 

8. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

#### 9. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

#### e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;  
 B. n = r;  
 C. z = r;  
 D. r = z;  
 E. z = n;
- 

#### 10. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
 B. Viene lanciata una NullPointerException.  
 C. Viene lanciata una ArrayIndexOutOfBoundsException.  
 D. null  
 E. Nessuna delle precedenti.
- 

#### 11. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

**12. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**13. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

**14. Dato il seguente codice:**

```

1. public interface I {
2.     int n = 0;
3. }

```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

**15. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Vehicle d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

```
Car d = new Car("Citroen", "C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

16. Data la seguente classe:

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 38**

1. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

2. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
    }
}
```

```

        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3
B. Fiat Opel Corsa Citroen C3
C. Citroen Citroen Citroen C3
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

7. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
    }
}

```

```

        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1**
  - B. 2**
  - C. 3**
  - D. 4**
  - E. Nessuna delle precedenti**
- 

#### 8. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.**
  - B. Viene lanciata una NullPointerException.**
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.**
  - D. null**
  - E. Nessuna delle precedenti.**
- 

#### 9. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;**
  - B. abstract int n = 0;**
  - C. protected int n = 0;**
  - D. public int n;**
  - E. Nessuna delle precedenti**
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
            try {

```

```

                System.out.print(1);
                p();
            }
            catch( Exception i ) {
                System.out.print(2);
            }
            catch( MyExc3 a ) {
            }
            finally {
                System.out.print(3);
                throw( new MyExc2() );
            }
        }
static void p()
throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}
}

```

- A. 145**
  - B. 145632**
  - C. Errore a tempo di compilazione**
  - D. 145623Exception in thread "main" MyExc2**
  - E. Nessuna delle precedenti**
- 

#### 11. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

**12. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

**13. Date le dichiarazioni:**

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

**e le dichiarazioni di variabile:**

```

Super n;
B1 r;
C2 z;

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

**14. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.

- B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

**15. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " " ;
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**16. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- 
- A. Fiat Opel Citroen C3**  
**B. Fiat Opel Corsa Citroen C3**  
**C. Citroen Citroen Citroen C3**  
**D. Errore a tempo di compilazione.**  
**E. Errore a tempo di esecuzione.**



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 39**

1. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
```

```
        System.out.print(3);
        throw( new MyExc3() );
    }
    catch( MyExc2 f ) {
        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356
- E. Nessuna delle precedenti

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
```

```

    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**  
**B. Fiat Opel Corsa Citroen C3**  
**C. Citroen Citroen Citroen C3**  
**D. Errore a tempo di compilazione.**  
**E. Errore a tempo di esecuzione.**
- 

**4. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**  
**B. Fiat Opel Corsa Citroen C3**

**C. Citroen Citroen Citroen C3**

- D. Errore a tempo di compilazione.**  
**E. Errore a tempo di esecuzione.**
- 

**5. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri**  
**B. Quando due metodi hanno lo stesso nome si ha overloading**  
**C. Non tutti i tipi di eccezioni estendono la classe Object**  
**D. Un array non possiede dei membri**  
**E. Due metodi possono differire per il tipo di ritorno**
- 

**6. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

9. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
}
```

```
Object o = new Z();
f(y); f(o);
}

A. XX
B. XZ
C. ZZ
D. Errore a tempo di compilazione
E. Errore a tempo di esecuzione
```

---

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }
```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

**12. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**13. Qual'è l'output di questo programma?**

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

**14. Date le dichiarazioni:**

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

**15. Quale output si ottiene invocando il metodo q?**

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
}

if(i3 == i4) {
    System.out.print(1);
} else {
    System.out.print(0);
}

}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

**16. Dato il seguente codice:**

```
1. public interface I {
2.     int n = 0;
3. }
```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 40**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }

public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
 B. Fiat Opel Corsa Citroen C3  
 C. Citroen Citroen Citroen C3  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

```

```

        public String id() {
            return producer + " " + model + " ";
        }
    }

    class Main {
        public static void main(String[] args) {
            Vehicle v = new Vehicle("Fiat","500");
            Vehicle c = new Car("Opel","Corsa");
            Car d = new Car("Citroen","C3");
            System.out.print(v.id());
            System.out.print(c.id());
            System.out.print(d.id());
        }
    }
}

```

- A. Fiat Opel Citroen C3  
 B. Fiat Opel Corsa Citroen C3  
 C. Citroen Citroen Citroen C3  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

5. Dato il seguente codice:

1. public interface I {
2. int n = 0;
3. }

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;  
 B. abstract int n = 0;  
 C. protected int n = 0;  
 D. public int n;  
 E. Nessuna delle precedenti
- 

6. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
}

```

- A. XX

- B. X Z
  - C. Z Z
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

8. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

9. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
        }  
        catch( MyExc1 v ) {  
            System.out.print(2);  
        }  
        catch( Exception d ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc2() );  
        }  
    }  
    static void q()  
        throws Exception {  
        try {  
            System.out.print(5);  
            if (true) throw( new MyExc3() );  
            else throw( new MyExc1() );  
        }  
        catch( MyExc1 z ) {  
            System.out.print(6);  
        }  
        finally {  
            System.out.print(7);  
        }  
    }  
}
```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args){  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }
```

```
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 f ) {  
            System.out.print(4);  
        }  
        catch( MyExc3 v ) {  
            System.out.print(5);  
        }  
        catch( Exception i ) {  
            System.out.print(6);  
            throw( new MyExc1() );  
        }  
    }  
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```
    }  
}
```

- A. 011
- B. 111
- C. 110
- D. 010
- E. 100

---

**16. Date le dichiarazioni:**

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 41**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

3. Dato il seguente codice:

```
1. public interface I {
2.     int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

4. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata **private**.
- B. Una classe interna non può essere dichiarata **final**.
- C. Una classe interna non può essere dichiarata **static**.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

7. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

public static void main(String[] args) {
    Star s1 = new Star();
    Sun s2 = new Sun();
    s1 = s2;
    s2 = null; /*QUI*/
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " " ;
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}
```

```

}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

#### 9. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

#### e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;**
  - B. n = r;**
  - C. z = r;**
  - D. r = z;**
  - E. z = n;**
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {

```

```

            }
        }
        static void m() {
            try {
                System.out.print(3);
                throw( new MyExc3() );
            }
            catch( MyExc2 f ) {
                System.out.print(4);
            }
            catch( MyExc3 v ) {
                System.out.print(5);
            }
            catch( Exception i ) {
            }
            finally {
                System.out.print(6);
                throw( new MyExc1() );
            }
        }
    }
}

```

- A. 135**
  - B. 1356Exception in thread "main" MyExc1**
  - C. Errore a tempo di compilazione**
  - D. 1356**
  - E. Nessuna delle precedenti**
- 

#### 11. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri**
  - B. Quando due metodi hanno lo stesso nome si ha overloading**
  - C. Non tutti i tipi di eccezioni estendono la classe Object**
  - D. Un array non possiede dei membri**
  - E. Due metodi possono differire per il tipo di ritorno**
- 

#### 12. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

### 13. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

### 14. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
- B. XZ

### C. ZZ

- D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

### 15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

### 16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

```

```
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 42**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;
```

```
Vehicle(String p, String m) {
    producer = p;
    model = m;
}

public String id() {
    return producer + " ";
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
}
```

```

void m(Integer i2, Object f2, Integer i3) {
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

4. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

5. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

6. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

8. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

---

## 9. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
}  
void g() {  
    Y y = new Y();  
    Object o = new Z();  
    f(y); f(o);  
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

## 10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
            finally {  
                System.out.print(3);  
                throw( new MyExc2() );  
            }  
        }  
        static void p()  
            throws Exception {  
            try {  
                System.out.print(4);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 f ) {  
            }  
            catch( MyExc3 b ) {  
            }  
        }  
    }
```

```
}  
catch( MyExc1 f ) {  
    System.out.print(5);  
}  
finally {  
    System.out.print(6);  
    throw( new MyExc3() );  
}  
}  
}
```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

## 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}  
  
    public static String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-

## 12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

A. 135

B. 1356Exception in thread "main" MyExc1

C. Errore a tempo di compilazione

D. 1356

E. Nessuna delle precedenti

---

## 13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
    }
}
```

```
catch( Exception d ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
    throw( new MyExc2() );
}
static void q()
    throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
finally {
    System.out.print(7);
}
}
```

A. 1574Exception in thread "main" MyExc2

B. 157

C. 157342

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

## 14. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

A. private void f( float i)

B. public int f( int j)

C. private void f( int j)

D. public static void f( int j)

E. Nessuna delle precedenti

---

## 15. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A.** q = (Error) u;
  - B.** q = (Error) m;
  - C.** m = (Boolean) q;
  - D.** m = (Boolean) u;
  - E.** Nessuno dei precedenti
- 

Super n;  
B1 r;  
C2 z;

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A.** r = n;
  - B.** n = r;
  - C.** z = r;
  - D.** r = z;
  - E.** z = n;
- 

**16. Date le dichiarazioni:**

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 43**

1. Qual è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
    }
}
```

```
        catch( Exception i ) {
        }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
```

```

        System.out.print(2);
    }
    catch( MyExc3 a ) {
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
static void p()
throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}

```

- A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
- 

**5. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?**

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1  
B. 2

C. 3

D. 4

E. Nessuna delle precedenti

---

**6. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

**7. Data la seguente classe:**

1. public class C {
2. public void f( int i){}
3. }
4. }

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)

- B.** public int f( int j)
  - C.** private void f( int j)
  - D.** public static void f( int j)
  - E. Nessuna delle precedenti**
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2**
  - B. 157**
  - C. 157342**
  - D. Errore a tempo di compilazione**
  - E. Nessuna delle precedenti**
- 

#### 9. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}

```

```

    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}

```

- A. XX**
  - B. XZ**
  - C. ZZ**
  - D. Errore a tempo di compilazione**
  - E. Errore a tempo di esecuzione**
- 

#### 10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
- B. Fiat Opel Corsa Citroen C3**
- C. Citroen Citroen Citroen C3**
- D. Errore a tempo di compilazione.**

E. Errore a tempo di esecuzione.

---

11. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 011
- B. 111
- C. 110
- D. 010
- E. 100

---

12. Date le dichiarazioni:

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;

E. z = n;

---

13. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

---

14. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

---

15. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

---

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";
```

```
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 44

1. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
    void g() {  
        Y y = new Y();  
        Object o = new Z();  
        f(y); f(o);  
    }  
}
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

2. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

3. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

4. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

5. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

6. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

7. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

8. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
- B. 2

C. 3

D. 4

E. Nessuna delle precedenti

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

A. Fiat Opel Citroen C3

B. Fiat Opel Corsa Citroen C3

C. Citroen Citroen Citroen C3

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

---

10. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {

```

```

        System.out.print(0);
    }
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 011**
  - B. 111**
  - C. 110**
  - D. 010**
  - E. 100**
- 

**11. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
- B. Fiat Opel Corsa Citroen C3**

**C. Citroen Citroen Citroen C3**

- D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

**12. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2**

- B. 157**

- C. 157342**

- D. Errore a tempo di compilazione**

- E. Nessuna delle precedenti**
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {

```

```

        System.out.print(1);
        m();
    }
    catch( MyExc2 g ) {
        System.out.print(2);
    }
    catch( MyExc3 w ) {
    }
}
static void m() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    catch( MyExc2 f ) {
        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}
}

throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

#### 14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
}

```

```

A. 145
B. 145632
C. Errore a tempo di compilazione
D. 145623Exception in thread "main" MyExc2
E. Nessuna delle precedenti

```

---

#### 15. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

#### e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;**  
**B. n = r;**  
**C. z = r;**  
**D. r = z;**  
**E. z = n;**
- 

#### 16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
}

```

```

public String id() {
    return producer + " ";
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
    }
}

```

```

Vehicle c = new Car("Opel","Corsa");
Car d = new Car("Citroen","C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 45**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
    try {
        System.out.print(5);
        if (true) throw( new MyExc3() );
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
    }
}
```

```

    }
    catch( MyExc2 g ) {
        System.out.print(2);
    }
    catch( MyExc3 w ) {
    }
}
static void m() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    catch( MyExc2 f ) {
        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 135  
B. 1356Exception in thread "main" MyExc1  
C. Errore a tempo di compilazione  
D. 1356  
E. Nessuna delle precedenti
- 

##### 5. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;  
B. q = (Error) m;  
C. m = (Boolean) q;  
D. m = (Boolean) u;  
E. Nessuno dei precedenti
- 

##### 6. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

public static void main(String[] args){
    Star s1 = new Star();
    Sun s2 = new Sun();
    s1 = s2;
    s2 = null; /*QUI*/
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

##### 7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " " ;
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
            finally {  
                System.out.print(3);  
                throw( new MyExc2() );  
            }  
        }  
        static void p()  
            throws Exception {  
                try {  
                    System.out.print(4);  
                    throw( new MyExc1() );  
                }  
                catch( MyExc2 f ) {  
                }  
                catch( MyExc3 b ) {  
                }  
                catch( MyExc1 f ) {  
                    System.out.print(5);  
                }  
            }  
        }
```

```
}  
finally {  
    System.out.print(6);  
    throw( new MyExc3() );  
}  
}  
}
```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(){  
    }  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

11. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

13. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

14. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011
- B. 111
- C. 110
- D. 010
- E. 100

15. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.**
- B. Viene lanciata una NullPointerException.**
- C. Viene lanciata una ArrayIndexOutOfBoundsException.**
- D. null**
- E. Nessuna delle precedenti.**

---

**16. Data la seguente classe:**

1. public class C {

2.       public void f( int i){}  
3.  
4. }

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)**
  - B. public int f( int j)**
  - C. private void f( int j)**
  - D. public static void f( int j)**
  - E. Nessuna delle precedenti**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 46**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;
```

```
Vehicle(String p, String m) {
    producer = p;
    model = m;
}

public String id() {
    return producer + " ";
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata **private**.
- B. Una classe interna non può essere dichiarata **final**.
- C. Una classe interna non può essere dichiarata **static**.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.

E. Nessuna delle precedenti

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
```

```
        System.out.print(1);
        p();
    }
    catch( Exception i ) {
        System.out.print(2);
    }
    catch( MyExc3 a ) {
    }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
static void p()
throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

---

6. Dato il seguente codice:

```
1.  public interface I {
2.      int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

9. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}

```

```

    }
}

class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

10. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
}

```

```

static void m() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    catch( MyExc2 f ) {
        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 135  
B. 1356Exception in thread "main" MyExc1  
C. Errore a tempo di compilazione  
D. 1356  
E. Nessuna delle precedenti
- 

## 12. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
B. Viene lanciata una NullPointerException.  
C. Viene lanciata una ArrayIndexOutOfBoundsException.  
D. null  
E. Nessuna delle precedenti.
- 

## 13. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;  
B. q = (Error) m;  
C. m = (Boolean) q;  
D. m = (Boolean) u;  
E. Nessuno dei precedenti
- 

## 14. Quanti oggetti *al massimo possono essere deallocated al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

## 15. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        }
    }
}

```

```
    } else {
        System.out.print(0);
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

16. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 47**

1. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

2. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
- 

- C. Viene lanciata una ArrayIndexOutOfBoundsException.

- D. null
  - E. Nessuna delle precedenti.
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

4. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-

5. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 f ) {  
            System.out.print(4);  
        }  
        catch( MyExc3 v ) {  
            System.out.print(5);  
        }  
        catch( Exception i ) {  
        }  
        finally {  
            System.out.print(6);  
            throw( new MyExc1() );  
        }  
    }  
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356

E. Nessuna delle precedenti

7. Quanti oggetti **al massimo possono essere deallocated al punto indicato?**

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
public static void main(String[] args){  
    Star s1 = new Star();  
    Sun s2 = new Sun();  
    s1 = s2;  
    s2 = null; /*QUI*/  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
        }  
        catch( MyExc1 v ) {  
            System.out.print(2);  
        }  
        catch( Exception d ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc2() );  
        }  
    }  
    static void q()  
        throws Exception {  
        try {  
            System.out.print(5);  
            if (true) throw( new MyExc3() );  
            else throw( new MyExc1() );  
        }
```

```

        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2**
- B. 157**
- C. 157342**
- D. Errore a tempo di compilazione**
- E. Nessuna delle precedenti**
- 

**9. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
- B. Fiat Opel Corsa Citroen C3**
- C. Citroen Citroen Citroen C3**
- D. Errore a tempo di compilazione.**
- E. Errore a tempo di esecuzione.**
- 

**10. Date le dichiarazioni:**

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;**
- B. n = r;**
- C. z = r;**
- D. r = z;**
- E. z = n;**
- 

**11. Dato il seguente codice:**

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;**
- B. abstract int n = 0;**
- C. protected int n = 0;**
- D. public int n;**
- E. Nessuna delle precedenti**
- 

**12. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri**
- B. Quando due metodi hanno lo stesso nome si ha overloading**
- C. Non tutti i tipi di eccezioni estendono la classe Object**
- D. Un array non possiede dei membri**
- E. Due metodi possono differire per il tipo di ritorno**
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

#### 14. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;**
  - B. q = (Error) m;**
  - C. m = (Boolean) q;**
  - D. m = (Boolean) u;**
  - E. Nessuno dei precedenti**
- 

#### 15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145**
  - B. 145632**
  - C. Errore a tempo di compilazione**
  - D. 145623Exception in thread "main" MyExc2**
  - E. Nessuna delle precedenti**
- 

#### 16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
}

```

```

public String id() {
    return producer + " ";
}
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
}

public String id() {
    return producer + " " + model + " ";
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
    }
}

```

```

Vehicle c = new Car("Opel", "Corsa");
Car d = new Car("Citroen", "C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}

```

- 
- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 48**

1. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

2. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

3. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
```

```
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
```

```

        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc2 f ) {
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc1 f ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
        throw( new MyExc3() );
    }
}

```

- A. 145**
- B. 145632**
- C. Errore a tempo di compilazione**
- D. 145623Exception in thread "main" MyExc2**
- E. Nessuna delle precedenti**
- 

## 5. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011**
- B. 111**
- C. 110**
- D. 010**
- E. 100**

## 6. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

### e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;**
  - B. n = r;**
  - C. z = r;**
  - D. r = z;**
  - E. z = n;**
- 

## 7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " " ;
    }
}
class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}
    public String id() {
        return producer + " " + model + " ";
    }
}
class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**

- B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

**8. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

**9. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {

```

```

            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2**
  - B. 157**
  - C. 157342**
  - D. Errore a tempo di compilazione**
  - E. Nessuna delle precedenti**
- 

**10. Date le dichiarazioni:**

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;**
  - B. q = (Error) m;**
  - C. m = (Boolean) q;**
  - D. m = (Boolean) u;**
  - E. Nessuno dei precedenti**
- 

**11. Data la seguente classe:**

```

1. public class C {
2.     public void f( int i){}
3.
4. }
```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

## 12. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

## 13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}
```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

## 14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356

E. Nessuna delle precedenti

---

15. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

16. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 49**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

2. Quale output si ottiene invocando il metodo q?

```
class D {
```

```
private Integer i1 = new Integer(0);
private Float f1 = new Float(1.5);
private Integer i4 = new Integer(20);
void q() {
    m(i1, f1, new Integer(20));
}
void m(Integer i2, Object f2, Integer i3) {
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " ";
    }
}
```

```

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}
    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata **private**.
  - B. Una classe interna non può essere dichiarata **final**.
  - C. Una classe interna non può essere dichiarata **static**.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;

- D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

7. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}   
  
    public static String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

9. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i ) {}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i )
- B. public int f( int j )
- C. private void f( int j )
- D. public static void f( int j )
- E. Nessuna delle precedenti

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
        }  
        catch( MyExc1 v ) {  
            System.out.print(2);  
        }  
        catch( Exception d ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc2() );  
        }  
    }  
    static void q()  
        throws Exception {  
        try {  
            System.out.print(5);  
            if (true) throw( new MyExc3() );  
            else throw( new MyExc1() );  
        }  
        catch( MyExc1 z ) {  
            System.out.print(6);  
        }  
        finally {  
            System.out.print(7);  
        }  
    }  
}
```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

11. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

**12. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

**13. Date le dichiarazioni:**

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

**e le dichiarazioni di variabile:**

```

Super n;
B1 r;
C2 z;

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;

E. z = n;

---

**14. Qual'è l'output di questo programma?**

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

**15. Quale output si ottiene invocando il metodo g?**

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

**16. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- 
- A. Fiat Opel Citroen C3**  
**B. Fiat Opel Corsa Citroen C3**  
**C. Citroen Citroen Citroen C3**  
**D. Errore a tempo di compilazione.**  
**E. Errore a tempo di esecuzione.**



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 50**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public static String id() {
        return producer + " ";
    }
}
class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public static String id() {
        return producer + " " + model + " ";
    }
}
class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Dato il seguente codice:

```
1. public interface I {
2.     int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

4. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

5. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 011
- B. 111

C. 110

D. 010

E. 100

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 f ) {  
            System.out.print(4);  
        }  
        catch( MyExc3 v ) {  
            System.out.print(5);  
        }  
        catch( Exception i ) {  
        }  
    finally {  
        System.out.print(6);  
        throw( new MyExc1() );  
    }  
    }  
}
```

A. 135

B. 1356Exception in thread "main" MyExc1

C. Errore a tempo di compilazione

D. 1356

E. Nessuna delle precedenti

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }
```

```

public String id() {
    return producer + " ";
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
}

public String id() {
    return producer + " " + model + " ";
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 8. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

#### 9. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

#### 10. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

#### 11. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

#### 12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

```

```

}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
 B. Fiat Opel Corsa Citroen C3  
 C. Citroen Citroen Citroen C3  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

13. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;  
 B. n = r;  
 C. z = r;  
 D. r = z;  
 E. z = n;
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }

```

```

public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
 B. 157  
 C. 157342  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

15. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

public static void main(String[] args) {
    Star s1 = new Star();
    Sun s2 = new Sun();
    s1 = s2;
    s2 = null; /*QUI*/
}

```

```
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

#### 16. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {
```

```
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 51**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Quale output si ottiene invocando il metodo g?

---

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

4. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;

E. Nessuno dei precedenti

---

6. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

A. 1
B. 2
C. 3
D. 4
E. Nessuna delle precedenti

```

---

8. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

A. Errore a tempo di compilazione per errata dichiarazione di un array.
B. Viene lanciata una NullPointerException.
C. Viene lanciata una ArrayIndexOutOfBoundsException.

```

**D. null**

**E. Nessuna delle precedenti.**

### 9. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

**A. 011**

**B. 111**

**C. 110**

**D. 010**

**E. 100**

### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {

```

```

            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

**A. 135**

**B. 1356Exception in thread "main" MyExc1**

**C. Errore a tempo di compilazione**

**D. 1356**

**E. Nessuna delle precedenti**

### 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

**12. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

**13. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading

- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

**14. Date le dichiarazioni:**

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

**e le dichiarazioni di variabile:**

```

Super n;
B1 r;
C2 z;

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

**15. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
    }
}

```

```
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}
}
```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

16. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)**
  - B. public int f( int j)**
  - C. private void f( int j)**
  - D. public static void f( int j)**
  - E. Nessuna delle precedenti**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 52**

1. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata `private`.
  - B. Una classe interna non può essere dichiarata `final`.
  - C. Una classe interna non può essere dichiarata `static`.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una `NullPointerException`.
  - C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
  - D. `null`
  - E. Nessuna delle precedenti.
- 

3. Quale output si ottiene invocando il metodo `q`?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

    }
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

4. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe `Object`
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

5. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `q = (Error) u;`

- B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

6. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

7. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
        }
```

```
        finally {  
            System.out.print(3);  
            throw( new MyExc2() );  
        }  
    }  
    static void p()  
    throws Exception {  
        try {  
            System.out.print(4);  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 f ) {  
        }  
        catch( MyExc3 b ) {  
        }  
        catch( MyExc1 f ) {  
            System.out.print(5);  
        }  
        finally {  
            System.out.print(6);  
            throw( new MyExc3() );  
        }  
    }  
}
```

---

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                q();  
            }  
            catch( MyExc1 v ) {  
                System.out.print(2);  
            }  
            catch( Exception d ) {  
                System.out.print(3);  
            }  
            finally {  
                System.out.print(4);  
                throw( new MyExc2() );  
            }  
        }  
        static void q()  
            throws Exception {  
            try {  
                System.out.print(5);  
            }
```

```

        if (true) throw( new MyExc3() );
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
 B. 157  
 C. 157342  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

**10. Quanti oggetti *al massimo possono essere deallocati al punto indicato?***

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args){
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. Nessuna delle precedenti
- 

**11. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
}

```

```

    public String id() {
        return producer + " ";
    }

    class Car extends Vehicle {
        Car(String p, String m) {super(p,m);}
        public String id() {
            return producer + " " + model + " ";
        }
    }

    class Main {
        public static void main(String[] args) {
            Vehicle v = new Vehicle("Fiat","500");
            Vehicle c = new Car("Opel","Corsa");
            Car d = new Car("Citroen","C3");
            System.out.print(v.id());
            System.out.print(c.id());
            System.out.print(d.id());
        }
    }
}

```

- A. Fiat Opel Citroen C3  
 B. Fiat Opel Corsa Citroen C3  
 C. Citroen Citroen Citroen C3  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

**12. Quale output si ottiene invocando il metodo g?**

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}

```

- A. XX  
 B. XZ  
 C. ZZ  
 D. Errore a tempo di compilazione  
 E. Errore a tempo di esecuzione
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {

```

```

    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**  
**B. Fiat Opel Corsa Citroen C3**  
**C. Citroen Citroen Citroen C3**  
**D. Errore a tempo di compilazione.**  
**E. Errore a tempo di esecuzione.**
- 

**15. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

Super n;  
B1 r;  
C2 z;

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

**16. Date le dichiarazioni:**

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 53**

1. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

3. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
    }
}
```

```

}
catch( Exception i ) {
    System.out.print(2);
}
catch( MyExc3 a ) {
}
finally {
    System.out.print(3);
    throw( new MyExc2() );
}
}
static void p()
throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}
}

```

**A. 145**

**B. 145632**

**C. Errore a tempo di compilazione**

**D. 145623Exception in thread "main" MyExc2**

**E. Nessuna delle precedenti**

## 5. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
}

```

**A. XX**

**B. XZ**

**C. ZZ**

**D. Errore a tempo di compilazione**

**E. Errore a tempo di esecuzione**

## 6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

**A. Fiat Opel Citroen C3**

**B. Fiat Opel Corsa Citroen C3**

**C. Citroen Citroen Citroen C3**

**D. Errore a tempo di compilazione.**

**E. Errore a tempo di esecuzione.**

## 7. Qual è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

**A. Errore a tempo di compilazione per errata dichiarazione di un array.**

**B. Viene lanciata una**

NullPointerException.

**C. Viene lanciata una**

ArrayIndexOutOfBoundsException.

**D. null**

**E. Nessuna delle precedenti.**

---

## 8. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

**A. 011**

**B. 111**

**C. 110**

**D. 010**

**E. 100**

---

## 9. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

**A. private void f( float i)**

**B. public int f( int j)**

**C. private void f( int j)**

**D. public static void f( int j)**

**E. Nessuna delle precedenti**

---

## 10. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

**A. final int n = 0;**

**B. abstract int n = 0;**

**C. protected int n = 0;**

**D. public int n;**

**E. Nessuna delle precedenti**

---

## 11. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri**
- B. Quando due metodi hanno lo stesso nome si ha overloading**
- C. Non tutti i tipi di eccezioni estendono la classe Object**
- D. Un array non possiede dei membri**
- E. Due metodi possono differire per il tipo di ritorno**

---

## 12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
        }  
        catch( MyExc1 v ) {  
            System.out.print(2);  
        }  
        catch( Exception d ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc2() );  
        }  
    }  
    static void q()  
        throws Exception {  
        try {  
            System.out.print(5);  
        }
```

```

        if (true) throw( new MyExc3() );
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 1574Exception in thread "main" MyExc2**  
**B. 157**  
**C. 157342**  
**D. Errore a tempo di compilazione**  
**E. Nessuna delle precedenti**
- 

### 13. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;**  
**B. q = (Error) m;**  
**C. m = (Boolean) q;**  
**D. m = (Boolean) u;**  
**E. Nessuno dei precedenti**
- 

### 14. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;**  
**B. n = r;**  
**C. z = r;**  
**D. r = z;**
- 

**E. z = n;**

---

### 15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

### 16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

```

```
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 54**

1. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}
```

```
}
```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " ";
    }
}
class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
}
```

```

public String id() {
    return producer + " " + model + " ";
}
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituita alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}

```

```

Car(String p, String m) {super(p,m);}

public String id() {
    return producer + " " + model + " ";
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {

```

```

        System.out.print(7);
    }
}

```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata **private**.
  - B. Una classe interna non può essere dichiarata **final**.
  - C. Una classe interna non può essere dichiarata **static**.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

8. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011
- B. 111
- C. 110

D. 010

E. 100

---

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

10. Quanti oggetti **al massimo possono essere deallocati al punto indicato?**

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
}

```

```

String name = "Sun";
double massIdx = 1.98910;

public static void main(String[] args) {
    Star s1 = new Star();
    Sun s2 = new Sun();
    s1 = s2;
    s2 = null; /*QUI*/
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

#### 11. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

#### 12. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

#### 13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 14. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;
- B. n = r;
- C. z = r;

- D.  $r = z;$
  - E.  $z = n;$
- 

**15. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha **overloading**
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

**16. Data la seguente classe:**

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 55**

1. Dato il seguente codice:

```
1. public interface I {
2.     int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
}
```

```
public static String id() {
    return producer + " ";
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
- 

E. Errore a tempo di esecuzione.

---

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

7. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

---

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(){  
    }  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

9. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
}  
void g() {  
    Y y = new Y();  
    Object o = new Z();  
    f(y); f(o);  
}
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 f ) {  
            System.out.print(4);  
        }  
        catch( MyExc3 v ) {  
            System.out.print(5);  
        }  
        catch( Exception i ) {  
            System.out.print(6);  
            throw( new MyExc1() );  
        }  
    }  
}
```

```
    }
}
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

11. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

12. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null;    /*QUI*/
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

13. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. **Errore a tempo di compilazione per errata dichiarazione di un array.**
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

14. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
```

```

        System.out.print(2);
    }
    catch( Exception d ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc2() );
    }
}
static void q()
    throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}

```

```

        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2**
- B. 157**
- C. 157342**
- D. Errore a tempo di compilazione**
- E. Nessuna delle precedenti**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 56**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

2. Dato il seguente codice:

```
1. public interface I {
2.     int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

4. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.

D. null

E. Nessuna delle precedenti.

- 
5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(){}  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

A. Fiat Opel Citroen C3

B. Fiat Opel Corsa Citroen C3

C. Citroen Opel Citroen C3

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

- 
6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
        }
```

```
}  
    catch( MyExc1 v ) {  
        System.out.print(2);  
    }  
    catch( Exception d ) {  
        System.out.print(3);  
    }  
    finally {  
        System.out.print(4);  
        throw( new MyExc2() );  
    }  
}  
static void q()  
    throws Exception {  
    try {  
        System.out.print(5);  
        if (true) throw( new MyExc3() );  
        else throw( new MyExc1() );  
    }  
    catch( MyExc1 z ) {  
        System.out.print(6);  
    }  
    finally {  
        System.out.print(7);  
    }  
}
```

A. 1574Exception in thread "main" MyExc2

B. 157

C. 157342

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

- 
7. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
public static void main(String[] args){  
    Star s1 = new Star();  
    Sun s2 = new Sun();  
    s1 = s2;  
    s2 = null; /*QUI*/  
}
```

A. 1

B. 2

C. 3

D. 4

## E. Nessuna delle precedenti

---

### 8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

A. 135

B. 1356Exception in thread "main" MyExc1

C. Errore a tempo di compilazione

D. 1356

E. Nessuna delle precedenti

---

### 9. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

---

### 10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

A. 145

B. 145632

C. Errore a tempo di compilazione

D. 145623Exception in thread "main" MyExc2

E. Nessuna delle precedenti

---

### 11. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
}
```

```

void m(Integer i2, Object f2, Integer i3) {
    if(i1 == i2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(f2 == f1) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " " ;
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. 011  
 B. 111  
 C. 110  
 D. 010  
 E. 100
- 

**12. Quale output si ottiene invocando il metodo g?**

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}

```

- A. X X  
 B. X Z  
 C. Z Z  
 D. Errore a tempo di compilazione  
 E. Errore a tempo di esecuzione
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

- A. Fiat Opel Citroen C3  
 B. Fiat Opel Corsa Citroen C3  
 C. Citroen Citroen Citroen C3  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

**14. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.  
 B. Una classe interna non può essere dichiarata final.  
 C. Una classe interna non può essere dichiarata static.  
 D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.  
 E. Nessuna delle precedenti
- 

**15. Data la seguente classe:**

1. public class C {
2. public void f( int i){}
3. }
4. }

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

**16. Date le dichiarazioni:**

Boolean m;  
Error q;

---

```
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 57

1. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

2. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata **private**.
- B. Una classe interna non può essere dichiarata **final**.
- C. Una classe interna non può essere dichiarata **static**.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

3. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha **overloading**
- C. Non tutti i tipi di eccezioni estendono la classe **Object**
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

4. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
}  
void g() {  
    Y y = new Y();  
    Object o = new Z();  
    f(y); f(o);  
}
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
            finally {  
                System.out.print(3);  
                throw( new MyExc2() );  
            }  
        }  
        static void p()
```

```

throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc2 f ) {
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc1 f ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
        throw( new MyExc3() );
    }
}

```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

6. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

7. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.

- D. null
  - E. Nessuna delle precedenti.
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3
B. Fiat Opel Corsa Citroen C3
C. Citroen Citroen Citroen C3
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

```

```

    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
 B. Fiat Opel Corsa Citroen C3  
 C. Citroen Citroen Citroen C3  
 D. Errore a tempo di compilazione.  
 E. Errore a tempo di esecuzione.
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
    }
}

```

```

        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3  

B. Fiat Opel Corsa Citroen C3  

C. Citroen Citroen Citroen C3  

D. Errore a tempo di compilazione.  

E. Errore a tempo di esecuzione.

```

---

11. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)  
 B. public int f( int j)  
 C. private void f( int j)  
 D. public static void f( int j)  
 E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
    }
}

```

```

        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

**13. Date le dichiarazioni:**

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;  
B. n = r;  
C. z = r;  
D. r = z;  
E. z = n;
- 

**14. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?**

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1**  
**B. 2**  
**C. 3**  
**D. 4**  
**E. Nessuna delle precedenti**
- 

**15. Quale output si ottiene invocando il metodo q?**

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011**  
**B. 111**  
**C. 110**  
**D. 010**  
**E. 100**
- 

**16. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
}

```

```
}

static void m() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    catch( MyExc2 f ) {
        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}
```

---

- A. 135**
- B. 1356Exception in thread "main" MyExc1**
- C. Errore a tempo di compilazione**
- D. 1356**
- E. Nessuna delle precedenti**



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 58**

1. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una `NullPointerException`.
  - C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
  - D. null
  - E. Nessuna delle precedenti.
- 

2. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. `private void f( float i)`
  - B. `public int f( int j)`
  - C. `private void f( int j)`
  - D. `public static void f( int j)`
  - E. Nessuna delle precedenti
- 

3. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}
```

```
class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

A. 1
B. 2
C. 3
D. 4
E. Nessuna delle precedenti
```

---

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
    }
}
```

```

        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**  
**B. Fiat Opel Corsa Citroen C3**  
**C. Citroen Citroen Citroen C3**  
**D. Errore a tempo di compilazione.**  
**E. Errore a tempo di esecuzione.**
- 

#### 5. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

#### e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;**  
**B. n = r;**  
**C. z = r;**  
**D. r = z;**  
**E. z = n;**
- 

#### 6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

```

```

        }
    }

    class Main {
        public static void main(String[] args) {
            Vehicle v = new Vehicle("Fiat", "500");
            Vehicle c = new Car("Opel", "Corsa");
            Car d = new Car("Citroen", "C3");
            System.out.print(v.id());
            System.out.print(c.id());
            System.out.print(d.id());
        }
    }
}

```

- A. Fiat Opel Citroen C3**  
**B. Fiat Opel Corsa Citroen C3**  
**C. Citroen Citroen Citroen C3**  
**D. Errore a tempo di compilazione.**  
**E. Errore a tempo di esecuzione.**
- 

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }

public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

**B. 1356Exception in thread "main" MyExc1**

**C. Errore a tempo di compilazione**

**D. 1356**

**E. Nessuna delle precedenti**

---

**8. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car() {}  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

**A. Fiat Opel Citroen C3**

**B. Fiat Opel Corsa Citroen C3**

**C. Citroen Citroen Citroen C3**

**D. Errore a tempo di compilazione.**

**E. Errore a tempo di esecuzione.**

---

**9. Quale output si ottiene invocando il metodo g?**

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}
```

```
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
    void g() {  
        Y y = new Y();  
        Object o = new Z();  
        f(y); f(o);  
    }  
}
```

**A. XX**

**B. XZ**

**C. ZZ**

**D. Errore a tempo di compilazione**

**E. Errore a tempo di esecuzione**

---

**10. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
            finally {  
                System.out.print(3);  
                throw( new MyExc2() );  
            }  
        }  
        static void p()  
        throws Exception {  
            try {  
                System.out.print(4);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 f ) {  
            }  
            catch( MyExc3 b ) {  
            }  
            catch( MyExc1 f ) {  
                System.out.print(5);  
            }  
            finally {  
                System.out.print(6);  
                throw( new MyExc3() );  
            }  
        }  
}
```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

## 11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

## 12. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.

- B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

## 13. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

## 14. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

## 15. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

**16. Date le dichiarazioni:**

---

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 59**

1. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
        }
    }
}
```

```
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

4. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        }
    }
}
```

```

    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

6. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

```

```

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args){
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

8. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

```

```

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}
}

```

```

public String id() {
    return producer + " " + model + " ";
}
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 135**
  - B. 1356Exception in thread "main" MyExc1**
  - C. Errore a tempo di compilazione**
  - D. 1356**
  - E. Nessuna delle precedenti**
- 

#### 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

#### 12. Dato il seguente codice:

1. public interface I {
2. int n = 0;
3. }

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

### 13. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

### 14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
    }
}
```

```
Car d = new Car("Citroen", "C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Opel Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

### 15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {

    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
                p();
            }
            catch( Exception i ) {
                System.out.print(2);
            }
            catch( MyExc3 a ) {
            }
            finally {
                System.out.print(3);
                throw( new MyExc2() );
            }
        }
        static void p()
            throws Exception {
                try {
                    System.out.print(4);
                    throw( new MyExc1() );
                }
                catch( MyExc2 f ) {
                }
                catch( MyExc3 b ) {
                }
                catch( MyExc1 f ) {
                    System.out.print(5);
                }
                finally {
                    System.out.print(6);
                    throw( new MyExc3() );
                }
            }
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2

E. Nessuna delle precedenti

---

16. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
}
```

```
void g() {  
    Y y = new Y();  
    Object o = new Z();  
    f(y); f(o);  
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 60**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}
```

```
class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

2. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
```

```
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args){
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1**
  - B. 2**
  - C. 3**
  - D. 4**
  - E. Nessuna delle precedenti**
- 

3. **Quale output si ottiene invocando il metodo q?**

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
    }  
}  
}
```

- A. 011
- B. 111
- C. 110
- D. 010
- E. 100

---

#### 4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
            finally {  
                System.out.print(3);  
                throw( new MyExc2() );  
            }  
        }  
        static void p()  
            throws Exception {  
                try {  
                    System.out.print(4);  
                    throw( new MyExc1() );  
                }  
                catch( MyExc2 f ) {  
                }  
                catch( MyExc3 b ) {  
                }  
                catch( MyExc1 f ) {  
                    System.out.print(5);  
                }  
                finally {  
                    System.out.print(6);  
                    throw( new MyExc3() );  
                }  
            }  
    }  
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

---

#### 5. Dato il seguente codice:

```
1.  public interface I {  
2.      int n = 0;  
3.  }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

---

#### 6. Data la seguente classe:

```
1.  public class C {  
2.      public void f( int i ) {}  
3.  
4.  }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i )
- B. public int f( int j )
- C. private void f( int j )
- D. public static void f( int j )
- E. Nessuna delle precedenti

---

#### 7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                q();  
            }  
            catch( MyExc1 v ) {  
                System.out.print(2);  
            }  
            catch( Exception d ) {  
                System.out.print(3);  
            }  
            finally {  
                System.out.print(4);  
                throw( new MyExc2() );  
            }  
        }  
        static void q()  
    }
```

```

        throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
finally {
    System.out.print(7);
}
}
}

```

- A. 1574Exception in thread "main" MyExc2**  
**B. 157**  
**C. 157342**  
**D. Errore a tempo di compilazione**  
**E. Nessuna delle precedenti**
- 

**8. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**

- B. Fiat Opel Corsa Citroen C3**  
**C. Citroen Citroen Citroen C3**  
**D. Errore a tempo di compilazione.**  
**E. Errore a tempo di esecuzione.**
- 

**9. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

**10. Date le dichiarazioni:**

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

**e le dichiarazioni di variabile:**

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

---

## 11. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

---

## 12. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
    void g() {  
        Y y = new Y();  
        Object o = new Z();  
        f(y); f(o);  
    }  
}
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione

## E. Errore a tempo di esecuzione

---

## 13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}  
  
    public static String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat", "500");  
        Vehicle c = new Car("Opel", "Corsa");  
        Car d = new Car("Citroen", "C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

## 14. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

---

## 15. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

16. Date le dichiarazioni:

```
Boolean m;  
Error q;
```

---

```
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 61**

1. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

2. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
```

```
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " " +
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
}

public String id() {
    return producer + " " + model + " ";
}
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
```

```

void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX  
B. XZ  
C. ZZ  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
B. Quando due metodi hanno lo stesso nome si ha overloading  
C. Non tutti i tipi di eccezioni estendono la classe Object  
D. Un array non possiede dei membri  
E. Due metodi possono differire per il tipo di ritorno
- 

6. Qual è l'output di questo codice?

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

#### 9. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

---

#### 10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " " ;
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}
    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3
B. Fiat Opel Corsa Citroen C3
C. Citroen Citroen Citroen C3
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

#### 11. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {

```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

12. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
    finally {
        System.out.print(3);
        throw( new MyExc2() );
    }
}

```

```

static void p()
    throws Exception {
    try {
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc2 f ) {
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc1 f ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
        throw( new MyExc3() );
    }
}

```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

15. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;

E. `z = n;`

---

16. Quanti oggetti *al massimo* possono essere deallocated al punto indicato?

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;
```

```
public static void main(String[] args){  
    Star s1 = new Star();  
    Sun s2 = new Sun();  
    s1 = s2;  
    s2 = null; /*QUI*/  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 62**

1. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
```

```
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

3. Data la seguente classe:

```
1.  public class C {
2.      public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

## 5. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]){
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.

- B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

## 6. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

## 7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3

```

- B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

8. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
```

```
    try {
        System.out.print(5);
        if (true) throw( new MyExc3() );
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " ";
    }
}
class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}
}
```

```

public String id() {
    return producer + " " + model + " ";
}
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**12. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?**

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

**13. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.

- C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

**14. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

**15. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object

- D. Un array non possiede dei membri**
  - E. Due metodi possono differire per il tipo di ritorno**
- 

**16. Date le dichiarazioni:**

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

**e le dichiarazioni di variabile:**

```
Super n;
```

---

```
B1 r;  
C2 z;
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;**
  - B. n = r;**
  - C. z = r;**
  - D. r = z;**
  - E. z = n;**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 63**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}
```

```
class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

2. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
```

```
} }
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
```

```

public static void main(String[] args) {
    Vehicle v = new Vehicle("Fiat", "500");
    Vehicle c = new Car("Opel", "Corsa");
    Car d = new Car("Citroen", "C3");
    System.out.print(v.id());
    System.out.print(c.id());
    System.out.print(d.id());
}
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

#### 5. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
        static void m() {
            try {
                System.out.print(3);
                throw( new MyExc3() );
            }
            catch( MyExc2 f ) {
                System.out.print(4);
            }
            catch( MyExc3 v ) {
                System.out.print(5);
            }
            catch( Exception i ) {
            }
        finally {

```

```

        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 135  
B. 1356Exception in thread "main" MyExc1  
C. Errore a tempo di compilazione  
D. 1356  
E. Nessuna delle precedenti
- 

7. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011  
B. 111  
C. 110  
D. 010  
E. 100
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {

```

```

            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
B. Quando due metodi hanno lo stesso nome si ha overloading  
C. Non tutti i tipi di eccezioni estendono la classe Object  
D. Un array non possiede dei membri  
E. Due metodi possono differire per il tipo di ritorno
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.  
B. Una classe interna non può essere dichiarata final.  
C. Una classe interna non può essere dichiarata static.

- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.  
E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
B. Viene lanciata una NullPointerException.  
C. Viene lanciata una ArrayIndexOutOfBoundsException.  
D. null  
E. Nessuna delle precedenti.
- 

12. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;  
B. abstract int n = 0;  
C. protected int n = 0;  
D. public int n;  
E. Nessuna delle precedenti
- 

13. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args){  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }  
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
  
    class Car extends Vehicle {  
        Car(String p, String m) {super(p,m);}  
  
        public static String id() {  
            return producer + " " + model + " ";  
        }  
    }  
  
    class Main {  
        public static void main(String[] args) {  
            Vehicle v = new Vehicle("Fiat","500");  
            Vehicle c = new Car("Opel","Corsa");  
            Car d = new Car("Citroen","C3");  
            System.out.print(v.id());  
            System.out.print(c.id());  
            System.out.print(d.id());  
        }  
    }  
}
```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

15. Date le dichiarazioni:

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

**e le dichiarazioni di variabile:**

Super n;  
B1 r;  
C2 z;

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

---

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

**16. Data la seguente classe:**



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 64

1. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

2. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

3. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;

- E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

5. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

```

        throw( new MyExc2() );
    }
}
static void p()
throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}
}

```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
    }
}

```

```

        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

**9. Quanti oggetti *al massimo possono essere deallocated al punto indicato?***

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

**10. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {

```

```

            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135  
B. 1356Exception in thread "main" MyExc1  
C. Errore a tempo di compilazione  
D. 1356  
E. Nessuna delle precedenti
- 

**11. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
B. Quando due metodi hanno lo stesso nome si ha overloading  
C. Non tutti i tipi di eccezioni estendono la classe Object  
D. Un array non possiede dei membri  
E. Due metodi possono differire per il tipo di ritorno
- 

**12. Quale output si ottiene invocando il metodo q?**

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

if(f2 == f1) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(i3 == i4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3

**D. Errore a tempo di compilazione.**

**E. Errore a tempo di esecuzione.**

---

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

**A. Fiat Opel Citroen C3**

**B. Fiat Opel Corsa Citroen C3**

**C. Citroen Citroen Citroen C3**

**D. Errore a tempo di compilazione.**

**E. Errore a tempo di esecuzione.**

---

**15. Quale output si ottiene invocando il metodo g?**

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}

```

```
void g() {  
    Y y = new Y();  
    Object o = new Z();  
    f(y); f(o);  
}
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

16. Data la seguente classe:

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 65**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
```

```
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 { }
```

```

public static void main(String [] argv) {
    try {
        System.out.print(1);
        m();
    }
    catch( MyExc2 g ) {
        System.out.print(2);
    }
    catch( MyExc3 w ) {
    }
}
static void m() {
    try {
        System.out.print(3);
        throw( new MyExc3() );
    }
    catch( MyExc2 f ) {
        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

4. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading

- C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3
B. Fiat Opel Corsa Citroen C3
C. Citroen Citroen Citroen C3
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

7. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

---

8. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args){  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(){}  
    Car(String p, String m) {super(p,m);}
```

```
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat", "500");  
        Vehicle c = new Car("Opel", "Corsa");  
        Car d = new Car("Citroen", "C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

10. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

---

11. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

---

12. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011**
  - B. 111**
  - C. 110**
  - D. 010**
  - E. 100**
- 

### 13. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}

```

- A. XX**
- B. XZ**
- C. ZZ**
- D. Errore a tempo di compilazione**
- E. Errore a tempo di esecuzione**

### 14. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;**
  - B. q = (Error) m;**
  - C. m = (Boolean) q;**
  - D. m = (Boolean) u;**
  - E. Nessuno dei precedenti**
- 

### 15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

```
    }
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

---

#### 16. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
```

```
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 66

1. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
class MyExcl extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
            finally {  
                System.out.print(3);  
                throw( new MyExc2() );  
            }  
        }  
        static void p()  
            throws Exception {  
            try {  
                System.out.print(4);  
                throw( new MyExcl() );  
            }  
            catch( MyExc2 f ) {  
            }  
            catch( MyExc3 b ) {  
            }  
        }  
    }
```

```
}  
catch( MyExcl f ) {  
    System.out.print(5);  
}  
finally {  
    System.out.print(6);  
    throw( new MyExc3() );  
}  
}  
}  
  
A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
```

3. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {
```

```

Car(String p, String m) {super(p,m);}

public String id() {
    return producer + " " + model + " ";
}

}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

7. Data la seguente classe:

1. public class C {
2. public void f( int i){}
3. }
4. }

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

8. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri

- B. Quando due metodi hanno lo stesso nome si ha overloading**
  - C. Non tutti i tipi di eccezioni estendono la classe Object**
  - D. Un array non possiede dei membri**
  - E. Due metodi possono differire per il tipo di ritorno**
- 

#### 9. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;**
  - B. n = r;**
  - C. z = r;**
  - D. r = z;**
  - E. z = n;**
- 

#### 10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
```

```
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}
```

- A. 135**
  - B. 1356Exception in thread "main" MyExc1**
  - C. Errore a tempo di compilazione**
  - D. 1356**
  - E. Nessuna delle precedenti**
- 

#### 11. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011**
  - B. 111**
  - C. 110**
  - D. 010**
  - E. 100**
- 

#### 12. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

**14. Quale output si ottiene invocando il metodo g?**

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

**15. Qual'è l'output di questo programma?**

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

**16. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {

```

```

        System.out.print(2);
    }
    catch( Exception d ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc2() );
    }
}
static void q()
    throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}

```

```

        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2**
- B. 157**
- C. 157342**
- D. Errore a tempo di compilazione**
- E. Nessuna delle precedenti**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 67

1. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

2. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}
```

```
}  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat", "500");  
        Vehicle c = new Car("Opel", "Corsa");  
        Car d = new Car("Citroen", "C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}   
  
    public static String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

6. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

8. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 011
- B. 111
- C. 110
- D. 010
- E. 100

9. Date le dichiarazioni:

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

---

10. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args){  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
}  
void g() {  
    Y y = new Y();  
    Object o = new Z();  
    f(y); f(o);  
}
```

- A. XX

B. X Z

C. Z Z

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends Exception { }  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
        }  
        catch( MyExc1 v ) {  
            System.out.print(2);  
        }  
        catch( Exception d ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc2() );  
        }  
    }  
    static void q()  
        throws Exception {  
        try {  
            System.out.print(5);  
            if (true) throw( new MyExc3() );  
            else throw( new MyExc1() );  
        }  
        catch( MyExc1 z ) {  
            System.out.print(6);  
        }  
        finally {  
            System.out.print(7);  
        }  
    }  
}
```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

13. Dato il seguente codice:

```
1.  public interface I {  
2.      int n = 0;  
3.  }
```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

**14. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

**15. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356
- E. Nessuna delle precedenti

**16. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " ";
    }
}
```

```
class Car extends Vehicle {  
    Car() {}  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");
```

```
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 68**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
```

```
String producer, model;

Vehicle(String p, String m) {
    producer = p;
    model = m;
}

public String id() {
    return producer + " ";
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

3. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}

class Y extends X {}

class Z extends Y {
```

```

void f(Z z) {
    System.out.println("Z");
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

4. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] args)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
}

```

```

        throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
finally {
    System.out.print(7);
}
}

```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

8. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.

- B. Viene lanciata una  
NullPointerException.
- C. Viene lanciata una  
ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

---

9. Date le dichiarazioni:

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;  
B. n = r;  
C. z = r;  
D. r = z;  
E. z = n;

---

10. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args){  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }  
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}  
  
A. 011  
B. 111  
C. 110  
D. 010  
E. 100
```

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            p();  
        }  
        catch( Exception i ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 a ) {  
        }  
        finally {  
            System.out.print(3);  
            throw( new MyExc2() );  
        }  
    }  
    static void p()  
        throws Exception {
```

```

try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}
}

```

- A. 145**  
**B. 145632**  
**C. Errore a tempo di compilazione**  
**D. 145623Exception in thread "main" MyExc2**  
**E. Nessuna delle precedenti**
- 

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**  
**B. Fiat Opel Corsa Citroen C3**  
**C. Citroen Citroen Citroen C3**  
**D. Errore a tempo di compilazione.**  
**E. Errore a tempo di esecuzione.**
- 

**14. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

**15. Data la seguente classe:**

1. public class C {
2. public void f( int i){}
3. }
4. }

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

**16. Date le dichiarazioni:**

Boolean m;  
Error q;

---

```
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 69**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
    try {
        System.out.print(5);
        if (true) throw( new MyExc3() );
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Dato il seguente codice:

```
1. public interface I {
2.     int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

4. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

if(i3 == i4) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

6. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

7. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

9. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends Exception { }  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
            finally {  
                System.out.print(3);  
                throw( new MyExc2() );  
            }  
        }  
        static void p()  
            throws Exception {  
            try {  
                System.out.print(4);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 f ) {  
            }  
            catch( MyExc3 b ) {  
            }  
        }  
    }
```

```
}  
    catch( MyExc1 f ) {  
        System.out.print(5);  
    }  
    finally {  
        System.out.print(6);  
        throw( new MyExc3() );  
    }  
}
```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc1 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 f ) {  
            System.out.print(4);  
        }  
        catch( MyExc3 v ) {  
            System.out.print(5);  
        }  
        catch( Exception i ) {  
        }  
        finally {  
            System.out.print(6);  
            throw( new MyExc1() );  
        }  
    }  
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356
- E. Nessuna delle precedenti

---

12. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

13. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

14. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- 
- A. Fiat Opel Citroen C3**  
**B. Fiat Opel Corsa Citroen C3**  
**C. Citroen Citroen Citroen C3**  
**D. Errore a tempo di compilazione.**  
**E. Errore a tempo di esecuzione.**



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 70**

1. Data la seguente classe:

1. public class C {
2.       public void f( int i){}
3. }
4. }

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
```

```
}  
catch( MyExc2 g ) {  
    System.out.print(2);  
}  
catch( MyExc3 w ) {  
}  
}  
static void m() {  
    try {  
        System.out.print(3);  
        throw( new MyExc3() );  
    }  
    catch( MyExc2 f ) {  
        System.out.print(4);  
    }  
    catch( MyExc3 v ) {  
        System.out.print(5);  
    }  
    catch( Exception i ) {  
    }  
    finally {  
        System.out.print(6);  
        throw( new MyExc1() );  
    }  
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
```

```

        System.out.print(2);
    }
    catch( Exception d ) {
        System.out.print(3);
    }
    finally {
        System.out.print(4);
        throw( new MyExc2() );
    }
}
static void q()
    throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
finally {
    System.out.print(7);
}
}
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 5. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;  
B. abstract int n = 0;  
C. protected int n = 0;  
D. public int n;  
E. Nessuna delle precedenti
- 

#### 6. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;  
B. n = r;  
C. z = r;  
D. r = z;  
E. z = n;
- 

#### 7. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
B. Viene lanciata una NullPointerException.  
C. Viene lanciata una ArrayIndexOutOfBoundsException.  
D. null  
E. Nessuna delle precedenti.
- 

#### 8. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
    }
}

```

```

Vehicle c = new Car("Opel", "Corsa");
Car d = new Car("Citroen", "C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

12. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri

- B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno

---

13. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args) {  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null;      /*QUI*/  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti

**14. Quale output si ottiene invocando il metodo q?**

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```
    if(i3 == i4) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione

D. 145623Exception in thread "main" MyExc2

E. Nessuna delle precedenti

---

16. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
}
```

```
}  
void g() {  
    Y y = new Y();  
    Object o = new Z();  
    f(y); f(o);  
}  
}
```

A. XX

B. XZ

C. ZZ

D. Errore a tempo di compilazione

E. Errore a tempo di esecuzione

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 71**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

2. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata **private**.
  - B. Una classe interna non può essere dichiarata **final**.
  - C. Una classe interna non può essere dichiarata **static**.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

3. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

4. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
```

```

        Object o = new Z();
        f(y); f(o);
    }
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

9. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;
```

```
public static void main(String[] args) {
    Star s1 = new Star();
    Sun s2 = new Sun();
    s1 = s2;
    s2 = null; /*QUI*/
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

11. Qual è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.**
- B. Viene lanciata una NullPointerException.**
- C. Viene lanciata una ArrayIndexOutOfBoundsException.**
- D. null**
- E. Nessuna delle precedenti.**
- 

## 12. Quale output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011**
- B. 111**
- C. 110**
- D. 010**
- E. 100**
- 

## 13. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

## e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;**
- B. n = r;**
- C. z = r;**
- D. r = z;**
- E. z = n;**
- 

## 14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
- B. Fiat Opel Corsa Citroen C3**
- C. Citroen Citroen Citroen C3**
- D. Errore a tempo di compilazione.**
- E. Errore a tempo di esecuzione.**
- 

## 15. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

16. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 72**

1. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

2. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
```

```
try {
    System.out.print(1);
    m();
}
catch( MyExc2 g ) {
    System.out.print(2);
}
catch( MyExc3 w ) {
}
}
static void m() {
try {
    System.out.print(3);
    throw( new MyExc3() );
}
catch( MyExc2 f ) {
    System.out.print(4);
}
catch( MyExc3 v ) {
    System.out.print(5);
}
catch( Exception i ) {
}
finally {
    System.out.print(6);
    throw( new MyExc1() );
}
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

---

5. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

---

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(){  
    }  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

```
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}  
  
    public static String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

---

8. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

10. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

11. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

12. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.

- C. Viene lanciata una  
    `ArrayIndexOutOfBoundsException`.  
D. null  
E. Nessuna delle precedenti.
- 

**13. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }

public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
- 

**14. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata  
    private.

- B. Una classe interna non può essere dichiarata  
    final.  
C. Una classe interna non può essere dichiarata  
    static.  
D. Una istanza di una classe interna può essere  
    creata solo all'interno della classe includente.  
E. Nessuna delle precedenti
- 

**15. Quale output si ottiene invocando il metodo q?**

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011  
B. 111  
C. 110  
D. 010  
E. 100
- 

**16. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }

public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
    }
}
```

```

}
catch( Exception d ) {
    System.out.print(3);
}
finally {
    System.out.print(4);
    throw( new MyExc2() );
}
}

static void q()
    throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
finally {
    System.out.print(7);
}
}

```

- A. 1574Exception in thread "main" MyExc2**
- B. 157**
- C. 157342**
- D. Errore a tempo di compilazione**
- E. Nessuna delle precedenti**
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 73

1. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
        }
    }
}
```

```

        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}

```

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {  
    static String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public static String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}  
  
    public static String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat", "500");  
        Vehicle c = new Car("Opel", "Corsa");  
        Car d = new Car("Citroen", "C3");
```

```

        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

**4. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?**

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. Nessuna delle precedenti
- 

**5. Date le dichiarazioni:**

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;

- B. n = r;  
C. z = r;  
D. r = z;  
E. z = n;
- 

**6. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

**7. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
}

```

```

public String id() {
    return producer + " ";
}
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
}

public String id() {
    return producer + " " + model + " ";
}
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

## 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
    }
}

```

```

        catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 135**
  - B. 1356Exception in thread "main" MyExc1**
  - C. Errore a tempo di compilazione**
  - D. 1356**
  - E. Nessuna delle precedenti**
- 

## 9. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX**
  - B. X Z**
  - C. Z Z**
  - D. Errore a tempo di compilazione**
  - E. Errore a tempo di esecuzione**
- 

## 10. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;**
- B. q = (Error) m;**
- C. m = (Boolean) q;**
- D. m = (Boolean) u;**

## E. Nessuno dei precedenti

---

### 11. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

### 12. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

### 13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends Exception {}  
public class D1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                p();  
            }  
            catch( Exception i ) {  
                System.out.print(2);  
            }  
            catch( MyExc3 a ) {  
            }  
            finally {  
                System.out.print(3);  
                throw( new MyExc2() );  
            }  
        }  
        static void p()  
            throws Exception {  
                try {
```

```
                    System.out.print(4);  
                    throw( new MyExc1() );  
                }  
                catch( MyExc2 f ) {  
                }  
                catch( MyExc3 b ) {  
                }  
                catch( MyExc1 f ) {  
                    System.out.print(5);  
                }  
                finally {  
                    System.out.print(6);  
                    throw( new MyExc3() );  
                }  
            }
```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

### 14. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

### 15. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i ){  
3.     }  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i )
  - B. public int f( int j )
  - C. private void f( int j )
  - D. public static void f( int j )
  - E. Nessuna delle precedenti
- 

### 16. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {}  
    if(i1 == i2) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(f2 == f1) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(i3 == i4) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 74**

1. Date le dichiarazioni:

```
class Super extends Object { ... }
class C2 extends Super { ... }
class B1 extends Super { ... }
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
```

```
        System.out.print(4);
        throw( new MyExc1() );
    }
    catch( MyExc2 f ) {
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc1 f ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
        throw( new MyExc3() );
    }
}
```

A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti

---

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}
```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
        Vehicle c = new Car("Opel", "Corsa");
        Car d = new Car("Citroen", "C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3

D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

---

5. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}

class Y extends X {}

class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}

void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

6. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.

- 
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti

---

8. Data la seguente classe:

```
1. public class C {  
2.     public void f( int i){}  
3.  
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

9. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```
public class C {  
    public static void main(String args[]) {  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

---

11. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc1 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc2 g ) {  
            System.out.print(2);  
        }  
        catch( MyExc3 w ) {  
        }  
    }  
    static void m() {  
        try {  
            System.out.print(3);  
            throw( new MyExc3() );  
        }  
        catch( MyExc2 f ) {  
            System.out.print(4);  
        }  
    }  
}
```

```

}
catch( MyExc3 v ) {
    System.out.print(5);
}
catch( Exception i ) {
}
finally {
    System.out.print(6);
    throw( new MyExc1() );
}
}

```

- A. 135  
B. 1356Exception in thread "main" MyExc1  
C. Errore a tempo di compilazione  
D. 1356  
E. Nessuna delle precedenti
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

**14. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

**15. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?**

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
}

```

```
String name = "Sun";
double massIdx = 1.98910;

public static void main(String[] args){
    Star s1 = new Star();
    Sun s2 = new Sun();
    s1 = s2;
    s2 = null;    /*QUI*/
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

**16. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 75**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

2. Quale output si ottiene invocando il metodo g?

```
class X {
```

```
void f(X x) {
    System.out.println("X");
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}
```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
}
```

```

        throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
finally {
    System.out.print(7);
}
}
}

```

- A. 1574Exception in thread "main" MyExc2  
B. 157  
C. 157342  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3

- C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

5. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;  
B. n = r;  
C. z = r;  
D. r = z;  
E. z = n;
- 

6. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
B. Viene lanciata una NullPointerException.  
C. Viene lanciata una ArrayIndexOutOfBoundsException.  
D. null  
E. Nessuna delle precedenti.
- 

7. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {

```

```

        return producer + " ";
    }

}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }

```

```

        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

#### 9. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

#### 10. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

#### 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

**12. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?**

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args){
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1**
  - B. 2**
  - C. 3**
  - D. 4**
  - E. Nessuna delle precedenti**
- 

**13. Data la seguente classe:**

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)**
  - B. public int f( int j)**
  - C. private void f( int j)**
  - D. public static void f( int j)**
  - E. Nessuna delle precedenti**
- 

**14. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.**
  - B. Una classe interna non può essere dichiarata final.**
  - C. Una classe interna non può essere dichiarata static.**
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.**
  - E. Nessuna delle precedenti**
- 

**15. Quale output si ottiene invocando il metodo q?**

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```
if(i3 == i4) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}  
}  
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

16. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 76**

1. Qual è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una `NullPointerException`.
- C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
- D. null
- E. Nessuna delle precedenti.

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
        }
    }
}
```

```
else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
finally {
    System.out.print(7);
}
}
}
A. 1574Exception in thread "main" MyExc2
B. 157
C. 157342
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```

3. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata `private`.
- B. Una classe interna non può essere dichiarata `final`.
- C. Una classe interna non può essere dichiarata `static`.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
```

```

Car() {}
Car(String p, String m) {super(p,m);}

public String id() {
    return producer + " " + model + " ";
}

}

```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145
  - B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

7. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

8. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
  - B. public int f( int j)
  - C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {

```

```

            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}

```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.

E. Errore a tempo di esecuzione.

- 
11. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args){  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

- 
12. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B. Quando due metodi hanno lo stesso nome si ha overloading
- C. Non tutti i tipi di eccezioni estendono la classe Object
- D. Un array non possiede dei membri
- E. Due metodi possono differire per il tipo di ritorno

- 
13. Quale output si ottiene invocando il metodo g?

```
class X {  
    void f(X x) {  
        System.out.println("X");  
    }  
}  
class Y extends X {}  
class Z extends Y {  
    void f(Z z) {  
        System.out.println("Z");  
    }  
    void g() {
```

```
        Y y = new Y();  
        Object o = new Z();  
        f(y); f(o);  
    }
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

- 
14. Date le dichiarazioni:

```
class Super extends Object {...}  
class C2 extends Super {...}  
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;  
B1 r;  
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
- B. n = r;
- C. z = r;
- D. r = z;
- E. z = n;

- 
15. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {  
        if(i1 == i2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(f2 == f1) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(i3 == i4) {  
            System.out.print(1);  
        }
```

```
    } else {
        System.out.print(0);
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

16. Dato il seguente codice:

---

```
1. public interface I {
2.     int n = 0;
3. }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 77**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri

- B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

3. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

4. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(i3 == i4) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
- 

- D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

6. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
        }
    }
}

```

```

        throw( new MyExc1() );
    }
    catch( MyExc2 f ) {
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc1 f ) {
        System.out.print(5);
    }
    finally {
        System.out.print(6);
        throw( new MyExc3() );
    }
}

```

- A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;  
B. q = (Error) m;  
C. m = (Boolean) q;  
D. m = (Boolean) u;  
E. Nessuno dei precedenti
- 

10. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

```

```

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
}

public static String id() {
    return producer + " " + model + " ";
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

11. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
}

```

- A. XX  
B. X Z  
C. ZZ  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

### 13. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

### 14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

### 15. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {

```

```
Star s1 = new Star();
Sun s2 = new Sun();
s1 = s2;
s2 = null; /*QUI*/
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

---

16. Data la seguente classe:

---

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 78**

1. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011
- B. 111
- C. 110
- D. 010
- E. 100

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
```

```
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}
```

- A. 1574Exception in thread "main" MyExc2
- B. 157
- C. 157342
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual'è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.

- C. Viene lanciata una  
ArrayIndexOutOfBoundsException.  
D. null  
E. Nessuna delle precedenti.

4. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}
```

- A. XX  
B. XZ  
C. ZZ  
D. Errore a tempo di compilazione  
E. Errore a tempo di esecuzione

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
```

```
public static void main(String[] args) {
    Vehicle v = new Vehicle("Fiat", "500");
    Vehicle c = new Car("Opel", "Corsa");
    Car d = new Car("Citroen", "C3");
    System.out.print(v.id());
    System.out.print(c.id());
    System.out.print(d.id());
}
```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.

6. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
B. Quando due metodi hanno lo stesso nome si ha overloading  
C. Non tutti i tipi di eccezioni estendono la classe Object  
D. Un array non possiede dei membri  
E. Due metodi possono differire per il tipo di ritorno

7. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.  
B. Una classe interna non può essere dichiarata final.  
C. Una classe interna non può essere dichiarata static.  
D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.  
E. Nessuna delle precedenti

8. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;

- B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

9. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
```

```
        System.out.print(4);
    }
    catch( MyExc3 v ) {
        System.out.print(5);
    }
    catch( Exception i ) {
    }
    finally {
        System.out.print(6);
        throw( new MyExc1() );
    }
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
}
```

```

        }
    }

    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
    finally {
        System.out.print(6);
        throw( new MyExc3() );
    }
}

```

**A. 145**

**B. 145632**

**C. Errore a tempo di compilazione**

**D. 145623Exception in thread "main" MyExc2**

**E. Nessuna delle precedenti**

---

### 13. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

**A. final int n = 0;**

**B. abstract int n = 0;**

**C. protected int n = 0;**

**D. public int n;**

**E. Nessuna delle precedenti**

---

### 14. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {

```

```

        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

**A. Fiat Opel Citroen C3**

**B. Fiat Opel Corsa Citroen C3**

**C. Citroen Citroen Citroen C3**

**D. Errore a tempo di compilazione.**

**E. Errore a tempo di esecuzione.**

---

### 15. Data la seguente classe:

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

**A. private void f( float i)**

**B. public int f( int j)**

**C. private void f( int j)**

**D. public static void f( int j)**

**E. Nessuna delle precedenti**

---

### 16. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {

```

```

        return producer + " ";
    }

}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");

```

```

        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 79**

1. Qual è l'output di questo programma?

```
public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una `NullPointerException`.
- C. Viene lanciata una `ArrayIndexOutOfBoundsException`.
- D. null
- E. Nessuna delle precedenti.

2. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `r = n;`
- B. `n = r;`
- C. `z = r;`
- D. `r = z;`
- E. `z = n;`

3. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

4. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}

class Y extends X {}
```

```

class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

**5. Quanti oggetti *al massimo possono essere deallocati al punto indicato?***

```

class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args){
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

**6. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
    }
}

```

```

        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

**7. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " " ;
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {

```

```

Vehicle v = new Vehicle("Fiat","500");
Vehicle c = new Car("Opel","Corsa");
Car d = new Car("Citroen","C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

#### 8. Qual è output si ottiene invocando il metodo q?

```

class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 011**
  - B. 111**
  - C. 110**
  - D. 010**
  - E. 100**
- 

#### 9. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;
}

```

```

Vehicle(String p, String m) {
    producer = p;
    model = m;
}

public String id() {
    return producer + " " ;
}

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

```

```

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3**
  - B. Fiat Opel Corsa Citroen C3**
  - C. Citroen Citroen Citroen C3**
  - D. Errore a tempo di compilazione.**
  - E. Errore a tempo di esecuzione.**
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
}

```

```

    }
    static void q()
        throws Exception {
    try {
        System.out.print(5);
        if (true) throw( new MyExc3() );
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
 B. 157  
 C. 157342  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

**11. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
 B. Quando due metodi hanno lo stesso nome si ha overloading  
 C. Non tutti i tipi di eccezioni estendono la classe Object  
 D. Un array non possiede dei membri  
 E. Due metodi possono differire per il tipo di ritorno
- 

**12. Data la seguente classe:**

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)  
 B. public int f( int j)  
 C. private void f( int j)  
 D. public static void f( int j)  
 E. Nessuna delle precedenti
- 

**13. Date le dichiarazioni:**

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.**

- A. q = (Error) u;  
 B. q = (Error) m;  
 C. m = (Boolean) q;  
 D. m = (Boolean) u;  
 E. Nessuno dei precedenti
- 

**14. Dato il seguente codice:**

```

1. public interface I {
2.     int n = 0;
3. }

```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;  
 B. abstract int n = 0;  
 C. protected int n = 0;  
 D. public int n;  
 E. Nessuna delle precedenti
- 

**15. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
    }
}

```

```
}

catch( MyExc2 f ) {
}

catch( MyExc3 b ) {
}

catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione

D. 145623Exception in thread "main" MyExc2

E. Nessuna delle precedenti

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 80**

1. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

2. Data la seguente classe:

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)

- C. private void f( int j)
  - D. public static void f( int j)
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- B. 145632
  - C. Errore a tempo di compilazione
  - D. 145623Exception in thread "main" MyExc2
  - E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

5. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car() {}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

```
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

7. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
  - B. Viene lanciata una NullPointerException.
  - C. Viene lanciata una ArrayIndexOutOfBoundsException.
  - D. null
  - E. Nessuna delle precedenti.
- 

8. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

9. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX
  - B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {
            System.out.print(7);
        }
    }
}

```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A.  $r = n$ ;
  - B.  $n = r$ ;
  - C.  $z = r$ ;
  - D.  $r = z$ ;
  - E.  $z = n$ ;
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

- A. 135
  - B. 1356Exception in thread "main" MyExc1
  - C. Errore a tempo di compilazione
  - D. 1356
  - E. Nessuna delle precedenti
- 

13. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;
```

```
Vehicle(String p, String m) {
    producer = p;
    model = m;
}

public static String id() {
    return producer + " ";
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata **private**.
  - B. Una classe interna non può essere dichiarata **final**.
  - C. Una classe interna non può essere dichiarata **static**.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

15. Dato il seguente codice:

```
1.  public interface I {
2.      int n = 0;
3.  }
```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;

- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

---

16. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";
```

```
double massIdx = 1.98910;  
  
public static void main(String[] args){  
    Star s1 = new Star();  
    Sun s2 = new Sun();  
    s1 = s2;  
    s2 = null; /*QUI*/  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 81**

1. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
  - B. Quando due metodi hanno lo stesso nome si ha overloading
  - C. Non tutti i tipi di eccezioni estendono la classe Object
  - D. Un array non possiede dei membri
  - E. Due metodi possono differire per il tipo di ritorno
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.
  - B. Una classe interna non può essere dichiarata final.
  - C. Una classe interna non può essere dichiarata static.
  - D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
    finally {
```

```
        System.out.print(4);
        throw( new MyExc2() );
    }
}
static void q()
    throws Exception {
try {
    System.out.print(5);
    if (true) throw( new MyExc3() );
    else throw( new MyExc1() );
}
catch( MyExc1 z ) {
    System.out.print(6);
}
finally {
    System.out.print(7);
}
}
```

- A. 1574Exception in thread "main" MyExc2
  - B. 157
  - C. 157342
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public String id() {
        return producer + " ";
    }
}
class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public String id() {
```

```

        return producer + " " + model + " ";
    }

}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

5. Date le dichiarazioni:

```

Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
  - B. q = (Error) m;
  - C. m = (Boolean) q;
  - D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

6. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

```

```

class Car extends Vehicle {
    Car(){}
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }

}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

A. Fiat Opel Citroen C3
B. Fiat Opel Corsa Citroen C3
C. Citroen Citroen Citroen C3
D. Errore a tempo di compilazione.
E. Errore a tempo di esecuzione.

```

---

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
    }
}

```

```

        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}

```

- A. 145  
 B. 145632  
 C. Errore a tempo di compilazione  
 D. 145623Exception in thread "main" MyExc2  
 E. Nessuna delle precedenti
- 

8. Data la seguente classe:

```

1. public class C {
2.     public void f( int i ) {}
3.
4. }

```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i )  
 B. public int f( int j )  
 C. private void f( int j )  
 D. public static void f( int j )  
 E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {

```

```

            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

A. 135  
 B. 1356Exception in thread "main" MyExc1  
 C. Errore a tempo di compilazione  
 D. 1356  
 E. Nessuna delle precedenti

---

10. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
 B. Viene lanciata una NullPointerException.  
 C. Viene lanciata una ArrayIndexOutOfBoundsException.  
 D. null  
 E. Nessuna delle precedenti.
- 

11. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
        System.out.println("Z");
    }
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}
}

```

- A. XX

- B. XZ
  - C. ZZ
  - D. Errore a tempo di compilazione
  - E. Errore a tempo di esecuzione
- 

12. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
  - B. Fiat Opel Corsa Citroen C3
  - C. Citroen Citroen Citroen C3
  - D. Errore a tempo di compilazione.
  - E. Errore a tempo di esecuzione.
- 

13. Quanti oggetti *al massimo possono essere deallocati al punto indicato?*

```
class Star {
    String name = "";
    double massIdx = 0;
}
```

```
class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```
class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}
```

e le dichiarazioni di variabile:

```
Super n;
B1 r;
C2 z;
```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;
  - B. n = r;
  - C. z = r;
  - D. r = z;
  - E. z = n;
- 

15. Dato il seguente codice:

```
1. public interface I {
2.     int n = 0;
3. }
```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

16. Quale output si ottiene invocando il metodo q?

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {}  
    if(i1 == i2) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(f2 == f1) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(i3 == i4) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 82**

1. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
```

```
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}
```

- A. 135
- B. 1356Exception in thread "main" MyExc1
- C. Errore a tempo di compilazione
- D. 1356
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception { }
```

```

try {
    System.out.print(1);
    p();
}
catch( Exception i ) {
    System.out.print(2);
}
catch( MyExc3 a ) {
}
finally {
    System.out.print(3);
    throw( new MyExc2() );
}
}
static void p()
throws Exception {
try {
    System.out.print(4);
    throw( new MyExc1() );
}
catch( MyExc2 f ) {
}
catch( MyExc3 b ) {
}
catch( MyExc1 f ) {
    System.out.print(5);
}
finally {
    System.out.print(6);
    throw( new MyExc3() );
}
}

```

- A. 145  
B. 145632  
C. Errore a tempo di compilazione  
D. 145623Exception in thread "main" MyExc2  
E. Nessuna delle precedenti
- 

4. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```

public class Vehicle {
    String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(){}

```

```

Car(String p, String m) {super(p,m);}

public String id() {
    return producer + " " + model + " ";
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}

```

- A. Fiat Opel Citroen C3  
B. Fiat Opel Corsa Citroen C3  
C. Citroen Citroen Citroen C3  
D. Errore a tempo di compilazione.  
E. Errore a tempo di esecuzione.
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends Exception { }
public class A1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            q();
        }
        catch( MyExc1 v ) {
            System.out.print(2);
        }
        catch( Exception d ) {
            System.out.print(3);
        }
        finally {
            System.out.print(4);
            throw( new MyExc2() );
        }
    }
    static void q()
        throws Exception {
        try {
            System.out.print(5);
            if (true) throw( new MyExc3() );
            else throw( new MyExc1() );
        }
        catch( MyExc1 z ) {
            System.out.print(6);
        }
        finally {

```

```

        System.out.print(7);
    }
}

```

- A. 1574Exception in thread "main" MyExc2  
 B. 157  
 C. 157342  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

6. Date le dichiarazioni:

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

e le dichiarazioni di variabile:

```

Super n;
B1 r;
C2 z;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. r = n;  
 B. n = r;  
 C. z = r;  
 D. r = z;  
 E. z = n;
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri  
 B. Quando due metodi hanno lo stesso nome si ha overloading  
 C. Non tutti i tipi di eccezioni estendono la classe Object  
 D. Un array non possiede dei membri  
 E. Due metodi possono differire per il tipo di ritorno
- 

8. Quale output si ottiene invocando il metodo g?

```

class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {

```

```

void f(Z z) {
    System.out.println("Z");
}
void g() {
    Y y = new Y();
    Object o = new Z();
    f(y); f(o);
}

```

- A. XX  
 B. XZ  
 C. ZZ  
 D. Errore a tempo di compilazione  
 E. Errore a tempo di esecuzione
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Una classe interna non può essere dichiarata private.  
 B. Una classe interna non può essere dichiarata final.  
 C. Una classe interna non può essere dichiarata static.  
 D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.  
 E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```

public class C {
    public static void main(String args[]) {
        C[] c = new C[10];
        System.out.println(c[6]);
    }
}

```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.  
 B. Viene lanciata una NullPointerException.  
 C. Viene lanciata una ArrayIndexOutOfBoundsException.  
 D. null  
 E. Nessuna delle precedenti.
- 

11. Dato il seguente codice:

```

1. public interface I {
2.     int n = 0;
3. }

```

Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?

- A. final int n = 0;
  - B. abstract int n = 0;
  - C. protected int n = 0;
  - D. public int n;
  - E. Nessuna delle precedenti
- 

12. Quale output si ottiene invocando il metodo q?

```
class D {
    private Integer i1 = new Integer(0);
    private Float f1 = new Float(1.5);
    private Integer i4 = new Integer(20);
    void q() {
        m(i1, f1, new Integer(20));
    }
    void m(Integer i2, Object f2, Integer i3) {
        if(i1 == i2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(f2 == f1) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(i3 == i4) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
- 

13. Date le dichiarazioni:

```
Boolean m;
Error q;
Object u;
q = new Error();
u = new Error();
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;

- D. m = (Boolean) u;
  - E. Nessuno dei precedenti
- 

14. Quanti oggetti ***al massimo possono essere deallocated al punto indicato?***

```
class Star {
    String name = "";
    double massIdx = 0;
}

class Sun extends Star {
    String name = "Sun";
    double massIdx = 1.98910;

    public static void main(String[] args) {
        Star s1 = new Star();
        Sun s2 = new Sun();
        s1 = s2;
        s2 = null; /*QUI*/
    }
}

A. 1
B. 2
C. 3
D. 4
E. Nessuna delle precedenti
```

---

15. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;

    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }

    public static String id() {
        return producer + " ";
    }
}

class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}

    public static String id() {
        return producer + " " + model + " ";
    }
}

class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat", "500");
    }
}
```

```
Vehicle c = new Car("Opel", "Corsa");
Car d = new Car("Citroen", "C3");
System.out.print(v.id());
System.out.print(c.id());
System.out.print(d.id());
}
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

```
1. public class C {
2.     public void f( int i){}
3.
4. }
```

Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?

- A. private void f( float i)
- B. public int f( int j)
- C. private void f( int j)
- D. public static void f( int j)
- E. Nessuna delle precedenti

---

16. Data la seguente classe:

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 83**

1. Date le dichiarazioni:

```
Boolean m;  
Error q;  
Object u;  
q = new Error();  
u = new Error();  
m = new Boolean(false);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. q = (Error) u;
- B. q = (Error) m;
- C. m = (Boolean) q;
- D. m = (Boolean) u;
- E. Nessuno dei precedenti

2. Quanti oggetti *al massimo* possono essere deallocati al punto indicato?

```
class Star {  
    String name = "";  
    double massIdx = 0;  
}  
  
class Sun extends Star {  
    String name = "Sun";  
    double massIdx = 1.98910;  
  
    public static void main(String[] args){  
        Star s1 = new Star();  
        Sun s2 = new Sun();  
        s1 = s2;  
        s2 = null; /*QUI*/  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. Nessuna delle precedenti

3. Qual è l'output di questo programma?

```
public class C {  
    public static void main(String args[]){  
        C[] c = new C[10];  
        System.out.println(c[6]);  
    }  
}
```

- A. Errore a tempo di compilazione per errata dichiarazione di un array.
- B. Viene lanciata una NullPointerException.
- C. Viene lanciata una ArrayIndexOutOfBoundsException.
- D. null
- E. Nessuna delle precedenti.

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends Exception {}  
public class A1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            q();  
        }  
        catch( MyExc1 v ) {  
            System.out.print(2);  
        }  
        catch( Exception d ) {  
            System.out.print(3);  
        }  
        finally {  
            System.out.print(4);  
            throw( new MyExc2() );  
        }  
    }  
    static void q()  
        throws Exception {  
        try {
```

```

        System.out.print(5);
        if (true) throw( new MyExc3() );
        else throw( new MyExc1() );
    }
    catch( MyExc1 z ) {
        System.out.print(6);
    }
    finally {
        System.out.print(7);
    }
}
}

```

- A. 1574Exception in thread "main" MyExc2**  
**B. 157**  
**C. 157342**  
**D. Errore a tempo di compilazione**  
**E. Nessuna delle precedenti**
- 

**5. Dire quale delle seguenti affermazioni è vera:**

- A. Due metodi non possono avere diverso nome ma lo stesso tipo di parametri**  
**B. Quando due metodi hanno lo stesso nome si ha overloading**  
**C. Non tutti i tipi di eccezioni estendono la classe Object**  
**D. Un array non possiede dei membri**  
**E. Due metodi possono differire per il tipo di ritorno**
- 

**6. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc1 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc2 g ) {
            System.out.print(2);
        }
        catch( MyExc3 w ) {
        }
    }
    static void m() {
        try {
            System.out.print(3);
            throw( new MyExc3() );
        }
        catch( MyExc2 f ) {
            System.out.print(4);
        }
    }
}

```

```

        }
        catch( MyExc3 v ) {
            System.out.print(5);
        }
        catch( Exception i ) {
        }
        finally {
            System.out.print(6);
            throw( new MyExc1() );
        }
    }
}

```

- A. 135**  
**B. 1356Exception in thread "main" MyExc1**  
**C. Errore a tempo di compilazione**  
**D. 1356**  
**E. Nessuna delle precedenti**
- 

**7. Data la seguente classe:**

```

1. public class C {
2.     public void f( int i){}
3.
4. }

```

**Quale dei seguenti metodi può essere legalmente aggiunto alla linea 3?**

- A. private void f( float i)**  
**B. public int f( int j)**  
**C. private void f( int j)**  
**D. public static void f( int j)**  
**E. Nessuna delle precedenti**
- 

**8. Date le dichiarazioni:**

```

class Super extends Object {...}
class C2 extends Super {...}
class B1 extends Super {...}

```

**e le dichiarazioni di variabile:**

```

Super n;
B1 r;
C2 z;

```

**indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.**

- A. r = n;**  
**B. n = r;**  
**C. z = r;**  
**D. r = z;**  
**E. z = n;**
-

## 9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends Exception { }
public class D1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            p();
        }
        catch( Exception i ) {
            System.out.print(2);
        }
        catch( MyExc3 a ) {
        }
        finally {
            System.out.print(3);
            throw( new MyExc2() );
        }
    }
    static void p()
        throws Exception {
        try {
            System.out.print(4);
            throw( new MyExc1() );
        }
        catch( MyExc2 f ) {
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc1 f ) {
            System.out.print(5);
        }
        finally {
            System.out.print(6);
            throw( new MyExc3() );
        }
    }
}
```

- A. 145
- B. 145632
- C. Errore a tempo di compilazione
- D. 145623Exception in thread "main" MyExc2
- E. Nessuna delle precedenti

## 10. Quale output si ottiene invocando il metodo g?

```
class X {
    void f(X x) {
        System.out.println("X");
    }
}
class Y extends X {}
class Z extends Y {
    void f(Z z) {
```

```
        System.out.println("Z");
    }
    void g() {
        Y y = new Y();
        Object o = new Z();
        f(y); f(o);
    }
}
```

- A. XX
- B. XZ
- C. ZZ
- D. Errore a tempo di compilazione
- E. Errore a tempo di esecuzione

## 11. Qual è il risultato della compilazione ed esecuzione del seguente programma?

```
public class Vehicle {
    static String producer, model;
    Vehicle(String p, String m) {
        producer = p;
        model = m;
    }
    public static String id() {
        return producer + " ";
    }
}
class Car extends Vehicle {
    Car(String p, String m) {super(p,m);}
    public static String id() {
        return producer + " " + model + " ";
    }
}
class Main {
    public static void main(String[] args) {
        Vehicle v = new Vehicle("Fiat","500");
        Vehicle c = new Car("Opel","Corsa");
        Car d = new Car("Citroen","C3");
        System.out.print(v.id());
        System.out.print(c.id());
        System.out.print(d.id());
    }
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

## 12. Dato il seguente codice:

```
1. public interface I {  
2.     int n = 0;  
3. }
```

**Quale delle seguenti linee può essere equivalentemente sostituite alla linea 2?**

- A. final int n = 0;
- B. abstract int n = 0;
- C. protected int n = 0;
- D. public int n;
- E. Nessuna delle precedenti

**13. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(){}  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

**14. Dire quale delle seguenti affermazioni è vera:**

- A. Una classe interna non può essere dichiarata private.
- B. Una classe interna non può essere dichiarata final.
- C. Una classe interna non può essere dichiarata static.
- D. Una istanza di una classe interna può essere creata solo all'interno della classe includente.
- E. Nessuna delle precedenti

**15. Qual è il risultato della compilazione ed esecuzione del seguente programma?**

```
public class Vehicle {  
    String producer, model;  
  
    Vehicle(String p, String m) {  
        producer = p;  
        model = m;  
    }  
  
    public String id() {  
        return producer + " ";  
    }  
}  
  
class Car extends Vehicle {  
    Car(String p, String m) {super(p,m);}  
  
    public String id() {  
        return producer + " " + model + " ";  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Vehicle v = new Vehicle("Fiat","500");  
        Vehicle c = new Car("Opel","Corsa");  
        Car d = new Car("Citroen","C3");  
        System.out.print(v.id());  
        System.out.print(c.id());  
        System.out.print(d.id());  
    }  
}
```

- A. Fiat Opel Citroen C3
- B. Fiat Opel Corsa Citroen C3
- C. Citroen Citroen Citroen C3
- D. Errore a tempo di compilazione.
- E. Errore a tempo di esecuzione.

**16. Quale output si ottiene invocando il metodo q?**

```
class D {  
    private Integer i1 = new Integer(0);  
    private Float f1 = new Float(1.5);  
    private Integer i4 = new Integer(20);  
    void q() {  
        m(i1, f1, new Integer(20));  
    }  
    void m(Integer i2, Object f2, Integer i3) {}  
    if(i1 == i2) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(f2 == f1) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
    if(i3 == i4) {  
        System.out.print(1);  
    } else {  
        System.out.print(0);  
    }  
}
```

- A. 011
  - B. 111
  - C. 110
  - D. 010
  - E. 100
-

**Prova n. 1**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 2**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 3**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 4**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 5**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 6**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 8**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 9**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 10**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 11**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 12**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 13**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**Prova n. 16**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

**LP1**

**Prova d'esame**

prof. Piero A. Bonatti

15 luglio 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  | A | B | C | D | E |
| 2  | A | B | C | D | E |
| 3  | A | B | C | D | E |
| 4  | A | B | C | D | E |
| 5  | A | B | C | D | E |
| 6  | A | B | C | D | E |
| 7  | A | B | C | D | E |
| 8  | A | B | C | D | E |
| 9  | A | B | C | D | E |
| 10 | A | B | C | D | E |
| 11 | A | B | C | D | E |
| 12 | A | B | C | D | E |
| 13 | A | B | C | D | E |
| 14 | A | B | C | D | E |
| 15 | A | B | C | D | E |
| 16 | A | B | C | D | E |

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   |   | ■ | ■ |   |
| 3  |  | ■ |   |   |   |   |   |
| 4  |  |   |   | ■ | ■ |   |   |
| 5  |  | ■ |   |   |   |   |   |
| 6  |  |   |   | ■ | ■ |   |   |
| 7  |  |   |   |   |   | ■ | ■ |
| 8  |  |   |   |   | ■ | ■ |   |
| 9  |  |   |   |   |   | ■ | ■ |
| 10 |  |   |   |   |   | ■ | ■ |
| 11 |  |   |   | ■ | ■ |   |   |
| 12 |  |   |   | ■ | ■ |   |   |
| 13 |  | ■ |   |   |   |   |   |
| 14 |  |   | ■ | ■ |   |   |   |
| 15 |  |   | ■ | ■ |   |   |   |
| 16 |  |   |   |   | ■ |   |   |

Risultato prova n. 1:

|    |  |   |   |   |   |   |  |
|----|--|---|---|---|---|---|--|
| 1  |  | ■ |   |   |   |   |  |
| 2  |  |   |   | ■ |   |   |  |
| 3  |  |   | ■ | ■ |   |   |  |
| 4  |  |   |   | ■ |   |   |  |
| 5  |  |   |   |   | ■ | ■ |  |
| 6  |  |   |   |   | ■ | ■ |  |
| 7  |  |   |   |   |   | ■ |  |
| 8  |  | ■ | ■ |   |   |   |  |
| 9  |  | ■ | ■ |   |   |   |  |
| 10 |  |   |   | ■ | ■ |   |  |
| 11 |  | ■ |   |   |   |   |  |
| 12 |  |   |   |   | ■ | ■ |  |
| 13 |  |   |   |   |   | ■ |  |
| 14 |  |   |   | ■ | ■ |   |  |
| 15 |  |   |   |   |   | ■ |  |
| 16 |  |   |   | ■ |   |   |  |

Risultato prova n. 2:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 3:

A 16x16 grid with red squares at the following coordinates:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (1, 5)
- (2, 5)
- (3, 5)
- (4, 5)
- (5, 5)
- (6, 5)
- (7, 5)
- (8, 5)
- (9, 5)
- (10, 5)
- (11, 5)
- (12, 5)
- (13, 5)
- (14, 5)
- (15, 5)
- (16, 5)
- (1, 6)
- (2, 6)
- (3, 6)
- (4, 6)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (1, 7)
- (2, 7)
- (3, 7)
- (4, 7)
- (5, 7)
- (6, 7)
- (7, 7)
- (8, 7)
- (9, 7)
- (10, 7)
- (11, 7)
- (12, 7)
- (13, 7)
- (14, 7)
- (15, 7)
- (16, 7)
- (1, 8)
- (2, 8)
- (3, 8)
- (4, 8)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (1, 9)
- (2, 9)
- (3, 9)
- (4, 9)
- (5, 9)
- (6, 9)
- (7, 9)
- (8, 9)
- (9, 9)
- (10, 9)
- (11, 9)
- (12, 9)
- (13, 9)
- (14, 9)
- (15, 9)
- (16, 9)
- (1, 10)
- (2, 10)
- (3, 10)
- (4, 10)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (1, 11)
- (2, 11)
- (3, 11)
- (4, 11)
- (5, 11)
- (6, 11)
- (7, 11)
- (8, 11)
- (9, 11)
- (10, 11)
- (11, 11)
- (12, 11)
- (13, 11)
- (14, 11)
- (15, 11)
- (16, 11)
- (1, 12)
- (2, 12)
- (3, 12)
- (4, 12)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (1, 13)
- (2, 13)
- (3, 13)
- (4, 13)
- (5, 13)
- (6, 13)
- (7, 13)
- (8, 13)
- (9, 13)
- (10, 13)
- (11, 13)
- (12, 13)
- (13, 13)
- (14, 13)
- (15, 13)
- (16, 13)
- (1, 14)
- (2, 14)
- (3, 14)
- (4, 14)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)
- (1, 15)
- (2, 15)
- (3, 15)
- (4, 15)
- (5, 15)
- (6, 15)
- (7, 15)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 15)
- (16, 15)
- (1, 16)
- (2, 16)
- (3, 16)
- (4, 16)
- (5, 16)
- (6, 16)
- (7, 16)
- (8, 16)
- (9, 16)
- (10, 16)
- (11, 16)
- (12, 16)
- (13, 16)
- (14, 16)
- (15, 16)
- (16, 16)

## Risultato prova n. 4:

|    |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|--|
| 1  |   | ■ |   |   |   |   |  |
| 2  |   |   | ■ |   |   |   |  |
| 3  |   |   |   | ■ | ■ |   |  |
| 4  |   |   |   |   | ■ |   |  |
| 5  |   |   |   |   |   | ■ |  |
| 6  |   |   |   | ■ |   |   |  |
| 7  |   |   |   |   | ■ |   |  |
| 8  |   | ■ |   |   |   |   |  |
| 9  |   |   |   |   | ■ |   |  |
| 10 |   |   |   |   |   | ■ |  |
| 11 | ■ |   |   |   |   |   |  |
| 12 | ■ |   |   |   |   |   |  |
| 13 |   |   |   |   |   | ■ |  |
| 14 |   |   |   | ■ |   |   |  |
| 15 |   |   |   |   | ■ |   |  |
| 16 |   |   | ■ |   |   |   |  |

Risultato prova n. 5:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 6:

|    |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|--|
| 1  |   | ■ |   |   |   |   |  |
| 2  |   |   |   |   | ■ | ■ |  |
| 3  |   |   |   | ■ | ■ |   |  |
| 4  |   |   |   | ■ | ■ |   |  |
| 5  |   | ■ | ■ |   |   |   |  |
| 6  |   |   |   |   | ■ | ■ |  |
| 7  |   |   |   |   | ■ | ■ |  |
| 8  | ■ | ■ |   |   |   |   |  |
| 9  |   |   |   |   | ■ | ■ |  |
| 10 |   |   | ■ | ■ |   |   |  |
| 11 | ■ | ■ |   |   |   |   |  |
| 12 |   |   |   | ■ | ■ |   |  |
| 13 |   |   |   | ■ | ■ |   |  |
| 14 |   |   |   |   | ■ | ■ |  |
| 15 | ■ | ■ |   |   |   |   |  |
| 16 |   |   |   |   | ■ |   |  |

Risultato prova n. 7:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 8:

|    |  |   |   |   |   |   |  |
|----|--|---|---|---|---|---|--|
| 1  |  | ■ |   |   |   |   |  |
| 2  |  |   |   | ■ |   |   |  |
| 3  |  | ■ |   |   |   |   |  |
| 4  |  |   |   | ■ |   |   |  |
| 5  |  |   |   |   | ■ |   |  |
| 6  |  |   | ■ |   |   |   |  |
| 7  |  |   |   |   |   | ■ |  |
| 8  |  |   | ■ |   |   |   |  |
| 9  |  |   |   |   |   | ■ |  |
| 10 |  |   |   |   | ■ |   |  |
| 11 |  | ■ |   |   |   |   |  |
| 12 |  | ■ |   |   |   |   |  |
| 13 |  |   |   | ■ |   |   |  |
| 14 |  |   |   | ■ |   |   |  |
| 15 |  |   |   |   |   | ■ |  |
| 16 |  |   |   |   | ■ |   |  |

Risultato prova n. 9:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 10:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 11:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  |   | ■ |   |   |   |   |
| 2  |  |   |   | ■ |   |   |   |
| 3  |  |   |   |   |   |   | ■ |
| 4  |  |   |   |   | ■ |   |   |
| 5  |  | ■ |   |   |   |   |   |
| 6  |  |   |   |   |   | ■ |   |
| 7  |  |   |   |   |   |   | ■ |
| 8  |  | ■ |   |   |   |   |   |
| 9  |  | ■ |   |   |   |   |   |
| 10 |  |   |   | ■ |   |   |   |
| 11 |  |   |   | ■ |   |   |   |
| 12 |  |   |   |   |   | ■ |   |
| 13 |  |   |   |   | ■ |   |   |
| 14 |  | ■ |   |   |   |   |   |
| 15 |  |   |   | ■ |   |   |   |
| 16 |  |   |   |   | ■ |   |   |

Risultato prova n. 12:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 13:

|    |  |  |  |  |  |  |    |
|----|--|--|--|--|--|--|----|
| 1  |  |  |  |  |  |  | 1  |
| 2  |  |  |  |  |  |  | 2  |
| 3  |  |  |  |  |  |  | 3  |
| 4  |  |  |  |  |  |  | 4  |
| 5  |  |  |  |  |  |  | 5  |
| 6  |  |  |  |  |  |  | 6  |
| 7  |  |  |  |  |  |  | 7  |
| 8  |  |  |  |  |  |  | 8  |
| 9  |  |  |  |  |  |  | 9  |
| 10 |  |  |  |  |  |  | 10 |
| 11 |  |  |  |  |  |  | 11 |
| 12 |  |  |  |  |  |  | 12 |
| 13 |  |  |  |  |  |  | 13 |
| 14 |  |  |  |  |  |  | 14 |
| 15 |  |  |  |  |  |  | 15 |
| 16 |  |  |  |  |  |  | 16 |

Risultato prova n. 14:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   |   |   |   | ■ |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  |   |   |   | ■ | ■ |   |   |
| 5  |   |   | ■ | ■ |   |   |   |
| 6  | ■ | ■ |   |   |   |   |   |
| 7  |   | ■ | ■ |   |   |   |   |
| 8  |   |   |   |   | ■ | ■ |   |
| 9  |   |   |   |   | ■ | ■ |   |
| 10 | ■ | ■ |   |   |   |   |   |
| 11 |   |   |   | ■ | ■ |   |   |
| 12 |   |   |   |   |   |   | ■ |
| 13 |   |   |   |   |   | ■ | ■ |
| 14 | ■ | ■ |   |   |   |   |   |
| 15 |   |   |   |   | ■ | ■ |   |
| 16 |   |   |   | ■ |   |   |   |

Risultato prova n. 15:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   |   | ■ | ■ |   |
| 3  |  |   |   |   | ■ | ■ |   |
| 4  |  | ■ | ■ |   |   |   |   |
| 5  |  | ■ | ■ |   |   |   |   |
| 6  |  |   | ■ | ■ |   |   |   |
| 7  |  |   |   | ■ | ■ |   |   |
| 8  |  |   |   |   |   | ■ | ■ |
| 9  |  |   |   |   | ■ | ■ |   |
| 10 |  |   |   |   |   | ■ | ■ |
| 11 |  | ■ | ■ |   |   |   |   |
| 12 |  |   | ■ | ■ |   |   |   |
| 13 |  |   |   |   | ■ | ■ |   |
| 14 |  |   |   |   |   | ■ | ■ |
| 15 |  |   |   | ■ | ■ |   |   |
| 16 |  |   |   | ■ | ■ |   |   |

Risultato prova n. 16:

|    |  |   |   |   |   |   |  |
|----|--|---|---|---|---|---|--|
| 1  |  | ■ |   |   |   |   |  |
| 2  |  |   |   |   |   | ■ |  |
| 3  |  | ■ |   |   |   |   |  |
| 4  |  |   |   | ■ |   |   |  |
| 5  |  |   |   |   |   | ■ |  |
| 6  |  |   |   |   | ■ |   |  |
| 7  |  |   | ■ |   |   |   |  |
| 8  |  |   |   |   |   | ■ |  |
| 9  |  | ■ |   |   |   |   |  |
| 10 |  | ■ |   |   |   |   |  |
| 11 |  |   |   | ■ |   |   |  |
| 12 |  |   | ■ |   |   |   |  |
| 13 |  |   |   | ■ |   |   |  |
| 14 |  |   |   |   | ■ |   |  |
| 15 |  |   |   |   | ■ |   |  |
| 16 |  |   |   | ■ |   |   |  |

Risultato prova n. 17:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   |   |   |   | ■ |
| 3  |  |   |   | ■ | ■ |   |   |
| 4  |  |   | ■ | ■ |   |   |   |
| 5  |  | ■ | ■ |   |   |   |   |
| 6  |  |   |   | ■ | ■ |   |   |
| 7  |  |   |   |   | ■ | ■ |   |
| 8  |  |   |   |   |   | ■ | ■ |
| 9  |  | ■ | ■ |   |   |   |   |
| 10 |  |   |   |   |   | ■ | ■ |
| 11 |  |   |   |   | ■ | ■ |   |
| 12 |  |   |   |   | ■ | ■ |   |
| 13 |  | ■ | ■ |   |   |   |   |
| 14 |  |   | ■ | ■ |   |   |   |
| 15 |  |   |   | ■ | ■ |   |   |
| 16 |  |   |   | ■ | ■ |   |   |

Risultato prova n. 18:

## Risultato prova n. 19:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 20:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 21:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 22:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  |   |   | ■ |   |   |
| 2  |   | ■ |   |   |   |
| 3  |   |   |   | ■ | ■ |
| 4  |   |   |   | ■ | ■ |
| 5  |   |   |   | ■ |   |
| 6  | ■ |   |   |   |   |
| 7  |   |   | ■ |   |   |
| 8  | ■ |   |   |   |   |
| 9  |   |   | ■ |   |   |
| 10 | ■ |   |   |   |   |
| 11 |   |   | ■ |   |   |
| 12 |   |   |   | ■ | ■ |
| 13 |   |   |   | ■ | ■ |
| 14 |   |   |   |   | ■ |
| 15 |   | ■ |   |   |   |
| 16 | ■ |   |   |   |   |

Risultato prova n. 23:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 24:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   |   | ■ | ■ |   |
| 3  |  |   | ■ | ■ |   |   |   |
| 4  |  |   |   | ■ |   |   |   |
| 5  |  | ■ | ■ |   |   |   |   |
| 6  |  | ■ | ■ |   |   |   |   |
| 7  |  | ■ | ■ |   |   |   |   |
| 8  |  |   |   |   |   | ■ | ■ |
| 9  |  |   |   | ■ | ■ |   |   |
| 10 |  |   |   | ■ | ■ |   |   |
| 11 |  |   |   |   | ■ | ■ |   |
| 12 |  |   |   | ■ | ■ |   |   |
| 13 |  |   |   | ■ | ■ |   |   |
| 14 |  |   |   |   | ■ | ■ |   |
| 15 |  |   |   |   | ■ | ■ |   |
| 16 |  |   |   |   |   | ■ | ■ |

Risultato prova n. 25:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  |   |   | ■ |   |   |
| 2  |   |   |   |   | ■ |
| 3  |   | ■ |   |   |   |
| 4  |   |   |   | ■ |   |
| 5  |   |   | ■ |   |   |
| 6  | ■ |   |   |   |   |
| 7  |   |   |   | ■ |   |
| 8  |   |   | ■ |   |   |
| 9  |   |   | ■ |   |   |
| 10 | ■ |   |   |   |   |
| 11 | ■ |   |   |   |   |
| 12 |   |   |   | ■ |   |
| 13 |   |   | ■ |   |   |
| 14 | ■ |   |   |   |   |
| 15 |   |   |   | ■ |   |
| 16 |   |   |   |   | ■ |

Risultato prova n. 26:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   |   |   |   | ■ |
| 3  |   |   |   |   | ■ | ■ |   |
| 4  | ■ |   |   |   |   |   |   |
| 5  |   |   |   |   | ■ | ■ |   |
| 6  |   |   | ■ | ■ |   |   |   |
| 7  |   |   |   | ■ | ■ |   |   |
| 8  |   |   | ■ | ■ |   |   |   |
| 9  |   |   |   | ■ | ■ |   |   |
| 10 |   |   |   |   |   |   | ■ |
| 11 |   |   |   |   | ■ |   |   |
| 12 | ■ | ■ |   |   |   |   |   |
| 13 | ■ | ■ |   |   |   |   |   |
| 14 |   |   |   | ■ | ■ |   |   |
| 15 |   |   |   |   | ■ |   |   |
| 16 |   |   |   |   | ■ |   |   |

Risultato prova n. 27:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  |   |   |   | ■ | ■ |   |   |
| 5  |   |   | ■ | ■ |   |   |   |
| 6  |   |   |   |   |   | ■ | ■ |
| 7  | ■ | ■ |   |   |   |   |   |
| 8  |   |   |   |   | ■ | ■ |   |
| 9  |   |   |   |   | ■ | ■ |   |
| 10 |   |   |   | ■ | ■ |   |   |
| 11 |   |   |   |   |   | ■ | ■ |
| 12 |   |   |   | ■ | ■ |   |   |
| 13 |   |   | ■ | ■ |   |   |   |
| 14 | ■ | ■ |   |   |   |   |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 |   |   |   |   |   | ■ | ■ |

Risultato prova n. 28:

A 16x16 grid with red squares at the following coordinates:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (1, 5)
- (2, 5)
- (3, 5)
- (4, 5)
- (5, 5)
- (6, 5)
- (7, 5)
- (8, 5)
- (9, 5)
- (10, 5)
- (11, 5)
- (12, 5)
- (13, 5)
- (14, 5)
- (15, 5)
- (16, 5)
- (1, 6)
- (2, 6)
- (3, 6)
- (4, 6)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (1, 7)
- (2, 7)
- (3, 7)
- (4, 7)
- (5, 7)
- (6, 7)
- (7, 7)
- (8, 7)
- (9, 7)
- (10, 7)
- (11, 7)
- (12, 7)
- (13, 7)
- (14, 7)
- (15, 7)
- (16, 7)
- (1, 8)
- (2, 8)
- (3, 8)
- (4, 8)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (1, 9)
- (2, 9)
- (3, 9)
- (4, 9)
- (5, 9)
- (6, 9)
- (7, 9)
- (8, 9)
- (9, 9)
- (10, 9)
- (11, 9)
- (12, 9)
- (13, 9)
- (14, 9)
- (15, 9)
- (16, 9)
- (1, 10)
- (2, 10)
- (3, 10)
- (4, 10)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (1, 11)
- (2, 11)
- (3, 11)
- (4, 11)
- (5, 11)
- (6, 11)
- (7, 11)
- (8, 11)
- (9, 11)
- (10, 11)
- (11, 11)
- (12, 11)
- (13, 11)
- (14, 11)
- (15, 11)
- (16, 11)
- (1, 12)
- (2, 12)
- (3, 12)
- (4, 12)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (1, 13)
- (2, 13)
- (3, 13)
- (4, 13)
- (5, 13)
- (6, 13)
- (7, 13)
- (8, 13)
- (9, 13)
- (10, 13)
- (11, 13)
- (12, 13)
- (13, 13)
- (14, 13)
- (15, 13)
- (16, 13)
- (1, 14)
- (2, 14)
- (3, 14)
- (4, 14)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)
- (1, 15)
- (2, 15)
- (3, 15)
- (4, 15)
- (5, 15)
- (6, 15)
- (7, 15)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 15)
- (16, 15)
- (1, 16)
- (2, 16)
- (3, 16)
- (4, 16)
- (5, 16)
- (6, 16)
- (7, 16)
- (8, 16)
- (9, 16)
- (10, 16)
- (11, 16)
- (12, 16)
- (13, 16)
- (14, 16)
- (15, 16)
- (16, 16)

## Risultato prova n. 29:

|    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|
| 1  |    |    | ■  |    |    |    |
| 2  |    |    |    |    | ■■ |    |
| 3  |    |    |    |    | ■■ |    |
| 4  |    |    |    |    |    | ■■ |
| 5  |    |    |    | ■■ |    |    |
| 6  |    | ■■ |    |    |    |    |
| 7  |    |    | ■■ |    |    |    |
| 8  |    |    | ■■ |    |    |    |
| 9  |    |    |    |    | ■■ |    |
| 10 | ■■ |    |    |    |    |    |
| 11 |    |    | ■■ |    |    |    |
| 12 | ■■ |    |    |    |    |    |
| 13 |    |    |    |    | ■■ |    |
| 14 |    |    |    | ■■ |    |    |
| 15 | ■■ |    |    |    |    |    |
| 16 | ■■ |    |    |    |    |    |

Risultato prova n. 30:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   |   | ■ | ■ |   |
| 3  |  |   |   |   |   | ■ | ■ |
| 4  |  |   |   | ■ | ■ |   |   |
| 5  |  | ■ |   |   |   |   |   |
| 6  |  |   | ■ | ■ |   |   |   |
| 7  |  |   |   |   | ■ | ■ |   |
| 8  |  | ■ |   |   |   |   |   |
| 9  |  |   | ■ | ■ |   |   |   |
| 10 |  |   |   |   | ■ | ■ |   |
| 11 |  |   |   | ■ | ■ |   |   |
| 12 |  |   |   | ■ | ■ |   |   |
| 13 |  |   |   |   |   | ■ | ■ |
| 14 |  |   |   | ■ | ■ |   |   |
| 15 |  | ■ |   |   |   |   |   |
| 16 |  |   |   |   |   | ■ | ■ |

Risultato prova n. 31:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   |   |   |   | ■ |
| 3  |   | ■ |   |   |   |   |   |
| 4  |   |   |   | ■ |   |   |   |
| 5  |   |   | ■ |   |   |   |   |
| 6  |   |   |   | ■ |   |   |   |
| 7  |   |   |   |   | ■ |   |   |
| 8  |   |   |   | ■ |   |   |   |
| 9  |   |   |   |   | ■ |   |   |
| 10 |   |   | ■ |   |   |   |   |
| 11 |   |   |   | ■ |   |   |   |
| 12 |   |   |   |   |   | ■ |   |
| 13 |   | ■ |   |   |   |   |   |
| 14 |   |   |   |   |   | ■ |   |
| 15 |   |   |   |   | ■ |   |   |
| 16 | ■ |   |   |   |   |   |   |

Risultato prova n. 32:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 33:

## Risultato prova n. 34:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   |   | ■ | ■ |   |
| 3  |  |   |   |   |   | ■ | ■ |
| 4  |  |   |   | ■ | ■ |   |   |
| 5  |  |   |   | ■ | ■ |   |   |
| 6  |  |   |   | ■ | ■ |   |   |
| 7  |  | ■ |   |   |   |   |   |
| 8  |  |   |   |   |   | ■ | ■ |
| 9  |  |   |   |   |   | ■ | ■ |
| 10 |  | ■ |   |   |   |   |   |
| 11 |  |   | ■ |   |   |   |   |
| 12 |  | ■ |   |   |   |   |   |
| 13 |  |   |   | ■ | ■ |   |   |
| 14 |  |   | ■ |   |   |   |   |
| 15 |  |   |   |   | ■ | ■ |   |
| 16 |  |   |   |   | ■ | ■ |   |

Risultato prova n. 35:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 36:

A 16x16 grid containing several red squares. The positions of the red squares are as follows:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (1, 5)
- (2, 5)
- (3, 5)
- (4, 5)
- (5, 5)
- (6, 5)
- (7, 5)
- (8, 5)
- (9, 5)
- (10, 5)
- (11, 5)
- (12, 5)
- (13, 5)
- (14, 5)
- (15, 5)
- (16, 5)
- (1, 6)
- (2, 6)
- (3, 6)
- (4, 6)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (1, 7)
- (2, 7)
- (3, 7)
- (4, 7)
- (5, 7)
- (6, 7)
- (7, 7)
- (8, 7)
- (9, 7)
- (10, 7)
- (11, 7)
- (12, 7)
- (13, 7)
- (14, 7)
- (15, 7)
- (16, 7)
- (1, 8)
- (2, 8)
- (3, 8)
- (4, 8)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (1, 9)
- (2, 9)
- (3, 9)
- (4, 9)
- (5, 9)
- (6, 9)
- (7, 9)
- (8, 9)
- (9, 9)
- (10, 9)
- (11, 9)
- (12, 9)
- (13, 9)
- (14, 9)
- (15, 9)
- (16, 9)
- (1, 10)
- (2, 10)
- (3, 10)
- (4, 10)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (1, 11)
- (2, 11)
- (3, 11)
- (4, 11)
- (5, 11)
- (6, 11)
- (7, 11)
- (8, 11)
- (9, 11)
- (10, 11)
- (11, 11)
- (12, 11)
- (13, 11)
- (14, 11)
- (15, 11)
- (16, 11)
- (1, 12)
- (2, 12)
- (3, 12)
- (4, 12)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (1, 13)
- (2, 13)
- (3, 13)
- (4, 13)
- (5, 13)
- (6, 13)
- (7, 13)
- (8, 13)
- (9, 13)
- (10, 13)
- (11, 13)
- (12, 13)
- (13, 13)
- (14, 13)
- (15, 13)
- (16, 13)
- (1, 14)
- (2, 14)
- (3, 14)
- (4, 14)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)
- (1, 15)
- (2, 15)
- (3, 15)
- (4, 15)
- (5, 15)
- (6, 15)
- (7, 15)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 15)
- (16, 15)
- (1, 16)
- (2, 16)
- (3, 16)
- (4, 16)
- (5, 16)
- (6, 16)
- (7, 16)
- (8, 16)
- (9, 16)
- (10, 16)
- (11, 16)
- (12, 16)
- (13, 16)
- (14, 16)
- (15, 16)
- (16, 16)

## Risultato prova n. 37:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 38:

|    |   |   |   |   |   |   |  |
|----|---|---|---|---|---|---|--|
| 1  |   | ■ |   |   |   |   |  |
| 2  |   |   |   | ■ |   |   |  |
| 3  |   |   |   |   | ■ |   |  |
| 4  |   |   | ■ |   |   |   |  |
| 5  |   |   |   |   |   | ■ |  |
| 6  |   |   |   | ■ |   |   |  |
| 7  |   |   |   |   |   | ■ |  |
| 8  |   |   | ■ |   |   |   |  |
| 9  |   |   |   |   | ■ |   |  |
| 10 |   |   |   |   |   | ■ |  |
| 11 | ■ |   |   |   |   |   |  |
| 12 |   |   |   | ■ |   |   |  |
| 13 |   |   |   |   | ■ |   |  |
| 14 | ■ |   |   |   |   |   |  |
| 15 |   |   |   | ■ |   |   |  |
| 16 | ■ |   |   |   |   |   |  |

Risultato prova n. 39:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 40:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 41:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  | ■ | ■ |   |   |   |   |   |
| 5  |   |   |   |   | ■ | ■ |   |
| 6  | ■ | ■ |   |   |   |   |   |
| 7  |   |   |   |   |   | ■ | ■ |
| 8  |   |   |   |   |   | ■ | ■ |
| 9  |   |   |   |   | ■ | ■ |   |
| 10 |   |   |   | ■ | ■ |   |   |
| 11 |   |   |   | ■ | ■ |   |   |
| 12 |   |   |   | ■ | ■ |   |   |
| 13 |   |   |   |   |   | ■ |   |
| 14 | ■ | ■ |   |   |   |   |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 |   | ■ |   |   |   |   |   |

Risultato prova n. 42:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 43:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 44:

## Risultato prova n. 45:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  |   |   | ■ |   |   |
| 2  |   |   |   | ■ |   |
| 3  |   |   |   |   | ■ |
| 4  |   |   |   |   | ■ |
| 5  |   |   | ■ |   |   |
| 6  | ■ |   |   |   |   |
| 7  |   |   | ■ |   |   |
| 8  |   |   |   |   | ■ |
| 9  |   |   |   | ■ |   |
| 10 |   | ■ |   |   |   |
| 11 |   |   | ■ |   |   |
| 12 |   |   |   | ■ |   |
| 13 | ■ |   |   |   |   |
| 14 | ■ |   |   |   |   |
| 15 |   |   | ■ |   |   |
| 16 | ■ |   |   |   |   |

Risultato prova n. 46:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 47:

|    |   |   |   |   |   |   |    |
|----|---|---|---|---|---|---|----|
| 1  |   |   |   |   |   |   | 1  |
| 2  |   |   |   |   |   |   | 2  |
| 3  |   | ■ |   |   |   |   | 3  |
| 4  |   |   |   | ■ | ■ |   | 4  |
| 5  |   |   |   | ■ | ■ |   | 5  |
| 6  |   |   | ■ | ■ |   |   | 6  |
| 7  |   |   |   |   | ■ | ■ | 7  |
| 8  |   |   | ■ | ■ |   |   | 8  |
| 9  |   |   |   |   |   | ■ | 9  |
| 10 |   | ■ | ■ |   |   |   | 10 |
| 11 |   | ■ | ■ |   |   |   | 11 |
| 12 |   |   |   |   | ■ | ■ | 12 |
| 13 |   |   |   | ■ | ■ |   | 13 |
| 14 |   |   |   | ■ | ■ |   | 14 |
| 15 |   |   |   |   |   | ■ | 15 |
| 16 | ■ | ■ |   |   |   |   | 16 |

Risultato prova n. 48:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 49:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 50:

|    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Risultato prova n. 51:

## Risultato prova n. 52:

A 16x16 grid with red squares at the following coordinates:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (1, 5)
- (2, 5)
- (3, 5)
- (4, 5)
- (5, 5)
- (6, 5)
- (7, 5)
- (8, 5)
- (9, 5)
- (10, 5)
- (11, 5)
- (12, 5)
- (13, 5)
- (14, 5)
- (15, 5)
- (16, 5)
- (1, 6)
- (2, 6)
- (3, 6)
- (4, 6)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (1, 7)
- (2, 7)
- (3, 7)
- (4, 7)
- (5, 7)
- (6, 7)
- (7, 7)
- (8, 7)
- (9, 7)
- (10, 7)
- (11, 7)
- (12, 7)
- (13, 7)
- (14, 7)
- (15, 7)
- (16, 7)
- (1, 8)
- (2, 8)
- (3, 8)
- (4, 8)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (1, 9)
- (2, 9)
- (3, 9)
- (4, 9)
- (5, 9)
- (6, 9)
- (7, 9)
- (8, 9)
- (9, 9)
- (10, 9)
- (11, 9)
- (12, 9)
- (13, 9)
- (14, 9)
- (15, 9)
- (16, 9)
- (1, 10)
- (2, 10)
- (3, 10)
- (4, 10)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (1, 11)
- (2, 11)
- (3, 11)
- (4, 11)
- (5, 11)
- (6, 11)
- (7, 11)
- (8, 11)
- (9, 11)
- (10, 11)
- (11, 11)
- (12, 11)
- (13, 11)
- (14, 11)
- (15, 11)
- (16, 11)
- (1, 12)
- (2, 12)
- (3, 12)
- (4, 12)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (1, 13)
- (2, 13)
- (3, 13)
- (4, 13)
- (5, 13)
- (6, 13)
- (7, 13)
- (8, 13)
- (9, 13)
- (10, 13)
- (11, 13)
- (12, 13)
- (13, 13)
- (14, 13)
- (15, 13)
- (16, 13)
- (1, 14)
- (2, 14)
- (3, 14)
- (4, 14)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)
- (1, 15)
- (2, 15)
- (3, 15)
- (4, 15)
- (5, 15)
- (6, 15)
- (7, 15)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 15)
- (16, 15)
- (1, 16)
- (2, 16)
- (3, 16)
- (4, 16)
- (5, 16)
- (6, 16)
- (7, 16)
- (8, 16)
- (9, 16)
- (10, 16)
- (11, 16)
- (12, 16)
- (13, 16)
- (14, 16)
- (15, 16)
- (16, 16)

## Risultato prova n. 53:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 54:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 55:

|    |   |   |   |   |   |
|----|---|---|---|---|---|
| 1  |   |   |   |   |   |
| 2  | ■ |   |   |   |   |
| 3  |   | ■ |   |   |   |
| 4  |   |   |   | ■ |   |
| 5  |   |   |   | ■ |   |
| 6  |   |   |   |   | ■ |
| 7  | ■ |   |   |   |   |
| 8  |   |   | ■ |   |   |
| 9  |   |   |   |   | ■ |
| 10 |   |   | ■ |   |   |
| 11 |   |   | ■ |   |   |
| 12 |   |   |   | ■ |   |
| 13 |   | ■ |   |   |   |
| 14 |   |   |   |   | ■ |
| 15 | ■ |   |   |   |   |
| 16 | ■ |   |   |   |   |

Risultato prova n. 56:

|    |    |   |    |    |    |    |  |
|----|----|---|----|----|----|----|--|
| 1  |    | ■ |    |    |    |    |  |
| 2  |    |   |    |    |    | ■■ |  |
| 3  |    |   |    |    |    | ■■ |  |
| 4  |    |   |    |    | ■■ |    |  |
| 5  |    |   |    | ■■ |    |    |  |
| 6  | ■■ |   |    |    |    |    |  |
| 7  |    |   |    |    | ■■ |    |  |
| 8  |    |   |    |    | ■■ |    |  |
| 9  |    |   | ■■ |    |    |    |  |
| 10 |    |   |    | ■■ |    |    |  |
| 11 | ■■ |   |    |    |    |    |  |
| 12 |    |   |    |    |    | ■■ |  |
| 13 |    |   | ■■ |    |    |    |  |
| 14 | ■■ |   |    |    |    |    |  |
| 15 |    |   |    | ■■ |    |    |  |
| 16 |    |   |    | ■■ |    |    |  |

Risultato prova n. 57:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 58:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   | ■ |   |   |   |   |   |
| 2  |   |   |   |   |   | ■ | ■ |
| 3  |   |   |   |   |   | ■ | ■ |
| 4  |   |   |   | ■ |   |   |   |
| 5  |   |   |   |   |   | ■ | ■ |
| 6  | ■ | ■ |   |   |   |   |   |
| 7  | ■ | ■ |   |   |   |   |   |
| 8  |   |   |   |   | ■ | ■ |   |
| 9  |   |   |   |   | ■ | ■ |   |
| 10 |   |   |   | ■ | ■ |   |   |
| 11 |   |   |   | ■ | ■ |   |   |
| 12 | ■ | ■ |   |   |   |   |   |
| 13 |   | ■ | ■ |   |   |   |   |
| 14 |   | ■ | ■ |   |   |   |   |
| 15 |   |   | ■ |   |   |   |   |
| 16 |   |   |   |   | ■ |   |   |

Risultato prova n. 59:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 60:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 61:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 62:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   |   | ■ | ■ |   |
| 3  |   |   |   |   | ■ | ■ |   |
| 4  |   |   |   | ■ |   |   |   |
| 5  | ■ | ■ |   |   |   |   |   |
| 6  |   |   |   | ■ | ■ |   |   |
| 7  |   |   |   | ■ | ■ |   |   |
| 8  |   |   |   |   |   | ■ | ■ |
| 9  |   |   |   |   |   | ■ | ■ |
| 10 |   |   |   |   |   | ■ | ■ |
| 11 |   |   |   |   | ■ |   |   |
| 12 | ■ | ■ |   |   |   |   |   |
| 13 | ■ | ■ |   |   |   |   |   |
| 14 |   |   |   | ■ | ■ |   |   |
| 15 |   |   | ■ | ■ |   |   |   |
| 16 | ■ | ■ |   |   |   |   |   |

Risultato prova n. 63:

A 16x16 grid with red squares at the following coordinates:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (1, 5)
- (2, 5)
- (3, 5)
- (4, 5)
- (5, 5)
- (6, 5)
- (7, 5)
- (8, 5)
- (9, 5)
- (10, 5)
- (11, 5)
- (12, 5)
- (13, 5)
- (14, 5)
- (15, 5)
- (16, 5)
- (1, 6)
- (2, 6)
- (3, 6)
- (4, 6)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (1, 7)
- (2, 7)
- (3, 7)
- (4, 7)
- (5, 7)
- (6, 7)
- (7, 7)
- (8, 7)
- (9, 7)
- (10, 7)
- (11, 7)
- (12, 7)
- (13, 7)
- (14, 7)
- (15, 7)
- (16, 7)
- (1, 8)
- (2, 8)
- (3, 8)
- (4, 8)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (1, 9)
- (2, 9)
- (3, 9)
- (4, 9)
- (5, 9)
- (6, 9)
- (7, 9)
- (8, 9)
- (9, 9)
- (10, 9)
- (11, 9)
- (12, 9)
- (13, 9)
- (14, 9)
- (15, 9)
- (16, 9)
- (1, 10)
- (2, 10)
- (3, 10)
- (4, 10)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (1, 11)
- (2, 11)
- (3, 11)
- (4, 11)
- (5, 11)
- (6, 11)
- (7, 11)
- (8, 11)
- (9, 11)
- (10, 11)
- (11, 11)
- (12, 11)
- (13, 11)
- (14, 11)
- (15, 11)
- (16, 11)
- (1, 12)
- (2, 12)
- (3, 12)
- (4, 12)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (1, 13)
- (2, 13)
- (3, 13)
- (4, 13)
- (5, 13)
- (6, 13)
- (7, 13)
- (8, 13)
- (9, 13)
- (10, 13)
- (11, 13)
- (12, 13)
- (13, 13)
- (14, 13)
- (15, 13)
- (16, 13)
- (1, 14)
- (2, 14)
- (3, 14)
- (4, 14)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)
- (1, 15)
- (2, 15)
- (3, 15)
- (4, 15)
- (5, 15)
- (6, 15)
- (7, 15)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 15)
- (16, 15)
- (1, 16)
- (2, 16)
- (3, 16)
- (4, 16)
- (5, 16)
- (6, 16)
- (7, 16)
- (8, 16)
- (9, 16)
- (10, 16)
- (11, 16)
- (12, 16)
- (13, 16)
- (14, 16)
- (15, 16)
- (16, 16)

## Risultato prova n. 64:

|    |   |  |   |   |   |   |  |
|----|---|--|---|---|---|---|--|
| 1  |   |  | ■ |   |   |   |  |
| 2  |   |  |   |   |   | ■ |  |
| 3  |   |  |   | ■ |   |   |  |
| 4  |   |  |   |   |   | ■ |  |
| 5  |   |  |   |   |   | ■ |  |
| 6  |   |  |   | ■ |   |   |  |
| 7  |   |  | ■ |   |   |   |  |
| 8  | ■ |  |   |   |   |   |  |
| 9  |   |  |   |   | ■ |   |  |
| 10 | ■ |  |   |   |   |   |  |
| 11 | ■ |  |   |   |   |   |  |
| 12 |   |  |   | ■ |   |   |  |
| 13 |   |  |   |   | ■ |   |  |
| 14 | ■ |  |   |   |   |   |  |
| 15 |   |  |   | ■ |   |   |  |
| 16 |   |  |   |   | ■ |   |  |

Risultato prova n. 65:

|    |  |   |   |   |   |   |   |
|----|--|---|---|---|---|---|---|
| 1  |  | ■ |   |   |   |   |   |
| 2  |  |   |   | ■ | ■ |   |   |
| 3  |  |   |   |   |   | ■ | ■ |
| 4  |  |   | ■ |   |   |   |   |
| 5  |  | ■ |   |   |   |   |   |
| 6  |  |   |   |   | ■ | ■ |   |
| 7  |  | ■ |   |   |   |   |   |
| 8  |  |   |   |   |   | ■ | ■ |
| 9  |  |   | ■ |   |   |   |   |
| 10 |  |   |   | ■ | ■ |   |   |
| 11 |  |   |   | ■ | ■ |   |   |
| 12 |  | ■ |   |   |   |   |   |
| 13 |  |   |   | ■ | ■ |   |   |
| 14 |  |   |   |   | ■ | ■ |   |
| 15 |  |   |   |   |   | ■ |   |
| 16 |  |   |   |   |   | ■ |   |

Risultato prova n. 66:

A 16x16 grid with red squares at the following coordinates:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (1, 5)
- (2, 5)
- (3, 5)
- (4, 5)
- (5, 5)
- (6, 5)
- (7, 5)
- (8, 5)
- (9, 5)
- (10, 5)
- (11, 5)
- (12, 5)
- (13, 5)
- (14, 5)
- (15, 5)
- (16, 5)
- (1, 6)
- (2, 6)
- (3, 6)
- (4, 6)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (1, 7)
- (2, 7)
- (3, 7)
- (4, 7)
- (5, 7)
- (6, 7)
- (7, 7)
- (8, 7)
- (9, 7)
- (10, 7)
- (11, 7)
- (12, 7)
- (13, 7)
- (14, 7)
- (15, 7)
- (16, 7)
- (1, 8)
- (2, 8)
- (3, 8)
- (4, 8)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (1, 9)
- (2, 9)
- (3, 9)
- (4, 9)
- (5, 9)
- (6, 9)
- (7, 9)
- (8, 9)
- (9, 9)
- (10, 9)
- (11, 9)
- (12, 9)
- (13, 9)
- (14, 9)
- (15, 9)
- (16, 9)
- (1, 10)
- (2, 10)
- (3, 10)
- (4, 10)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (1, 11)
- (2, 11)
- (3, 11)
- (4, 11)
- (5, 11)
- (6, 11)
- (7, 11)
- (8, 11)
- (9, 11)
- (10, 11)
- (11, 11)
- (12, 11)
- (13, 11)
- (14, 11)
- (15, 11)
- (16, 11)
- (1, 12)
- (2, 12)
- (3, 12)
- (4, 12)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (1, 13)
- (2, 13)
- (3, 13)
- (4, 13)
- (5, 13)
- (6, 13)
- (7, 13)
- (8, 13)
- (9, 13)
- (10, 13)
- (11, 13)
- (12, 13)
- (13, 13)
- (14, 13)
- (15, 13)
- (16, 13)
- (1, 14)
- (2, 14)
- (3, 14)
- (4, 14)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)
- (1, 15)
- (2, 15)
- (3, 15)
- (4, 15)
- (5, 15)
- (6, 15)
- (7, 15)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 15)
- (16, 15)
- (1, 16)
- (2, 16)
- (3, 16)
- (4, 16)
- (5, 16)
- (6, 16)
- (7, 16)
- (8, 16)
- (9, 16)
- (10, 16)
- (11, 16)
- (12, 16)
- (13, 16)
- (14, 16)
- (15, 16)
- (16, 16)

## Risultato prova n. 67:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 68:

## Risultato prova n. 69:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 70:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 71:

A 16x16 grid with red squares at the following coordinates:

- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)

## Risultato prova n. 72:

|    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Risultato prova n. 73:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   | ■ | ■ |   |   |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  |   |   |   |   |   | ■ | ■ |
| 5  |   |   |   |   |   | ■ | ■ |
| 6  | ■ | ■ |   |   |   |   |   |
| 7  |   |   |   |   |   |   | ■ |
| 8  | ■ | ■ |   |   |   |   |   |
| 9  | ■ | ■ |   |   |   |   |   |
| 10 |   |   |   |   | ■ | ■ |   |
| 11 |   |   |   | ■ | ■ |   |   |
| 12 |   |   |   | ■ | ■ |   |   |
| 13 |   |   |   |   |   | ■ | ■ |
| 14 |   |   | ■ | ■ |   |   |   |
| 15 | ■ | ■ |   |   |   |   |   |
| 16 |   |   |   |   |   | ■ | ■ |

Risultato prova n. 74:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 75:

A 16x16 grid containing several red squares. The squares are located at the following coordinates (row, column): (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1), (9, 1), (10, 1), (11, 1), (12, 1), (13, 1), (14, 1), (15, 1), and (16, 1). There are also additional red squares at (1, 2), (2, 2), (3, 2), (4, 2), (5, 2), (6, 2), (7, 2), (8, 2), (9, 2), (10, 2), (11, 2), (12, 2), (13, 2), (14, 2), (15, 2), and (16, 2).

## Risultato prova n. 76:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 77:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 78:

|    |  |  |  |  |  |  |  |
|----|--|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |  |

Risultato prova n. 79:

|    |  |  |  |  |  |
|----|--|--|--|--|--|
| 1  |  |  |  |  |  |
| 2  |  |  |  |  |  |
| 3  |  |  |  |  |  |
| 4  |  |  |  |  |  |
| 5  |  |  |  |  |  |
| 6  |  |  |  |  |  |
| 7  |  |  |  |  |  |
| 8  |  |  |  |  |  |
| 9  |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |

Risultato prova n. 80:

## Risultato prova n. 81:

|    |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|
| 1  |   |   | ■ |   |   |   |   |
| 2  |   |   |   | ■ | ■ |   |   |
| 3  |   |   |   | ■ | ■ |   |   |
| 4  |   |   |   |   |   | ■ |   |
| 5  |   |   |   |   |   |   | ■ |
| 6  |   |   | ■ |   |   |   |   |
| 7  |   |   |   |   |   |   | ■ |
| 8  |   |   |   |   | ■ | ■ |   |
| 9  |   |   |   |   |   |   | ■ |
| 10 |   |   |   |   | ■ | ■ |   |
| 11 | ■ | ■ |   |   |   |   |   |
| 12 |   |   |   | ■ | ■ |   |   |
| 13 | ■ | ■ |   |   |   |   |   |
| 14 | ■ | ■ |   |   |   |   |   |
| 15 |   |   |   | ■ | ■ |   |   |
| 16 | ■ | ■ |   |   |   |   |   |

Risultato prova n. 82:

|    |  |  |  |  |  |  |
|----|--|--|--|--|--|--|
| 1  |  |  |  |  |  |  |
| 2  |  |  |  |  |  |  |
| 3  |  |  |  |  |  |  |
| 4  |  |  |  |  |  |  |
| 5  |  |  |  |  |  |  |
| 6  |  |  |  |  |  |  |
| 7  |  |  |  |  |  |  |
| 8  |  |  |  |  |  |  |
| 9  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| 16 |  |  |  |  |  |  |

Risultato prova n. 83:



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 1**

1. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception { }
```

```

try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

5. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
E. Nessuna delle precedenti.
- 

6. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

7. Qual'è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

8. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}

```

```

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {

```

```

Boolean b1 = new Boolean(true);
if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;
- 

14. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
    }
}

```

```

c1 = null;
c3 = c1;
/* QUI */
return;
}
}
A. 1
B. 2
C. 3
D. 4
E. 6

```

---

15. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
        }
    }
}

```

```
        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 2**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    }
}
```

```
/* QUI */
return;
}

A. 1
B. 2
C. 3
D. 4
E. 6
```

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
- B. 1

- C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

6. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}
```

```
class House implements Building{  
    String type = "Residential";
```

```

int age;
public int getAge() {
    return age;
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 10. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {

```

```

protected void f(Z z) {
    System.out.print("Z");
}
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 11. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

#### 12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

14. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
- 

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {

```

```
        throw( new MyExc1() ) ;
    }
}

static void m()
throws Exception {
    try {
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}
```

- 
- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 3**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
```

```
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

4. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
}
```

```

public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

6. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

9. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
                m();
            }
```

```
}  
catch( Exception s ) {  
    System.out.print(2);  
    throw( new MyExc1() );  
}  
catch( MyExc2 z ) {  
    System.out.print(3);  
}  
finally {  
    throw( new MyExc1() );  
}  
}  
static void m()  
throws Exception {  
try {  
}  
catch( Exception h ) {  
}  
finally {  
    System.out.print(4);  
}  
}  
}
```

---

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

12. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

**14. Dato un file contenente il seguente codice:**

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

**15. Qual è l'output di questo programma?**

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

16. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
}
```

```
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 4**

1. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;
```

```

Coffee() {}
Coffee(int ml) {
    this.ml = ml;
}
public static void main(String args[]) {
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata `abstract` o `final`
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

6. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125

- B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new O.I()`.
- E. Nessuna delle precedenti.

---

9. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0

D. null

E. Nessuna delle precedenti

---

11. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 z ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc1() );  
            }  
        }  
        static void m()  
        throws Exception {  
            try {  
            }  
            catch( Exception h ) {  
            }  
            finally {  
                System.out.print(4);  
            }  
        }  
    }
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1

- C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

14. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

15. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

16. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
          Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {
```

```
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- 
- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 5**

1. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

2. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
} }
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

3. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
```

```

        super(6);
        System.out.print(5);
    }
}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

---

9. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
- B. 12222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

11. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
E. Nessuna delle precedenti.

---

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6
- 

13. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- 
- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 6**

1. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- ```
    }
}

A. 12Exception in thread "main" MyExc3
B. 1222222... (ciclo infinito)
C. 12
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
```

```

}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

6. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `x = (Error []) y;`
  - B. `x = y;`
  - C. `y = m;`
  - D. `x = m;`
  - E. `m = (Object []) x;`
- 

9. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
        }
    }
}
```

```
        throw( new MyExc1() );
    }
    catch( MyExc2 z ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
    try {
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

14. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

15. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{

```

```
Mammal() {  
    super(6);  
    System.out.print(5);  
}  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 7**

1. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
```

```
public static void main(String[] args) {
    X x = new Z();
    Y y = (Y)x;
    Z z = (Z)y;
    z.f(x); z.f(y); z.f(z);
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- B. 101
  - C. 100
  - D. 011
  - E. 010
- 

4. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

5. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {
```

```
        }  
    }  
}  
static void p() {  
    try {  
        System.out.print(1);  
        throw( new MyExc3() );  
    }  
    catch( MyExc3 y ) {  
        System.out.print(2);  
        throw( new MyExc3() );  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc3 b ) {  
        }  
        catch( MyExc2 j ) {  
        }  
        catch( Exception v ) {  
            System.out.print(2);  
        }  
        finally {  
            throw( new MyExc2() );  
        }  
    }  
    static void m()  
        throws Exception {  
        try {  
        }  
        catch( Exception g ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

8. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

9. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

A. 1
B. 2
C. 3
D. 4
E. 6
```

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

12. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

15. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

16. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}
```

```
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- ```
    }  
  
A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
```
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 8**

1. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

2. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
-

---

4. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 z ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc1() );  
            }  
        }  
        static void m()  
            throws Exception {  
                try {  
                }  
                catch( Exception h ) {  
                }  
                finally {  
                    System.out.print(4);  
                }  
            }  
    }
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
}  
public static void main(String args[]) {  
    House h = new House();  
}
```

```

        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

11. Qual è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 9**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}
```

```
class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
 B. XYZ  
 C. ZZZ  
 D. Errore a tempo di compilazione.  
 E. Nessuna delle precedenti.
- 

5. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
    }
}

```

- z.f(x); z.f(y); z.f(z);  
 }  
 A. XXX  
 B. XYZ  
 C. ZZZ  
 D. Errore a tempo di compilazione.  
 E. Nessuna delle precedenti.
- 

6. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. 6
- 

7. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
 B. x = y;  
 C. y = m;  
 D. x = m;  
 E. m = (Object []) x;
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        } catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        } catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 9. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 10. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;  
B. g = (Boolean) s;  
C. b = (Integer) g;  
D. g = (Boolean) b;  
E. Nessuno dei precedenti
- 

#### 11. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
E. Nessuna delle precedenti.
- 

#### 12. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}

```

```

}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

14. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {

```

```

            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {

```

```
try {  
}  
catch( Exception g ) {  
}  
finally {  
    System.out.print(3);  
}  
}  
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 10**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
```

```
} }
```

```
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
```

```

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;

```
E. m = (Object []) x;
```

---

6. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {

```

```

        System.out.print(3);
    }
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

8. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
E. Nessuna delle precedenti.
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
B. Una classe può essere dichiarata abstract o final  
C. Due metodi non possono differire per il tipo di ritorno  
D. Un array non possiede dei membri  
E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] args) {
        try {
            p();
        }
    }
}

```

```

        catch( MyExc2 r ) {
    }
}
static void p() {
    try {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}

```

---

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

**13. Qual è l'output di questo codice?**

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**14. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?**

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

**15. Qual è l'output di questo codice?**

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

16. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 11**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

```
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

---

4. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building {  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0

D. null

E. Nessuna delle precedenti

---

6. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal {  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]) {  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3

- D. 4  
E. 6
- 

8. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
B. Una classe può essere dichiarata **abstract** o **final**  
C. Due metodi non possono differire per il tipo di ritorno  
D. Un array non possiede dei membri  
E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();
```

```
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( Exception s ) {  
            System.out.print(2);  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 z ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc1() );  
        }  
    }  
    static void m()  
        throws Exception {  
        try {  
        }  
        catch( Exception h ) {  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)

E. Nessuna delle precedenti

---

12. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

13. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

14. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

---

15. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
    finally {

```

```
        throw( new MyExc2() );
    }
}

static void m()
throws Exception {
    try {
    }
    catch( Exception g ) {
    }
    finally {
        System.out.print(3);
    }
}
```

- }
- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 12**

1. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
    }}
```

```
}
```

```
static void p() {
    try {
        System.out.print(1);
        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

---

5. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

6. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( Exception s ) {  
            System.out.print(2);  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 z ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc1() );  
        }  
    }  
    static void m()  
        throws Exception {  
        try {
```

```

    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {

```

```

        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

11. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
-

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
```

```
public static void main(String args[]) {
    System.out.print(1);
    Mammal m = new Mammal();
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

15. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-

16. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 13**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
```

```
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
```

```

Boolean b3 = new Boolean(true);
s7 = s6;
q(b3, s2, s7, s6);
}
void q(Boolean b2, String s3, String s4,
        Object s5) {
    Boolean b1 = new Boolean(true);
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

5. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

6. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;
}

```

```

Coffee() {}
Coffee(int ml){
    this.ml = ml;
}
public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}
A. 1
B. 2
C. 3
D. 4
E. 6

```

---

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

9. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

10. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

11. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;

E. m = (Object []) x;

---

12. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 z ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc1() );  
            }  
        }  
        static void m()  
            throws Exception {
```

```
        try {  
        }  
        catch( Exception h ) {  
        }  
        finally {  
            System.out.print(4);  
        }  
    }
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

15. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 14**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
```

```
public static void main(String args[]) {
    System.out.print(1);
    Mammal m = new Mammal();
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

3. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata `abstract` o `final`
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

4. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

5. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

6. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

---

E. Nessuna delle precedenti.

---

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
    }
}

```

```

        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {

```

```

        System.out.print(1);
        m();
    }
    catch( Exception s ) {
        System.out.print(2);
        throw( new MyExc1() );
    }
    catch( MyExc2 z ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
    try {
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

### 13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
            try {

```

```

        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

### 14. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;  
B. g = (Boolean) s;  
C. b = (Integer) g;  
D. g = (Boolean) b;  
E. Nessuno dei precedenti
- 

### 15. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;
- 

### 16. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 15**

1. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

3. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

4. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

5. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
}
public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
}

```

```

        return;
    }
}

A. 1
B. 2
C. 3
D. 4
E. 6

```

---

7. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

8. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

```

```

    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `x = (Error []) y;`
  - B. `x = y;`
  - C. `y = m;`
  - D. `x = m;`
  - E. `m = (Object []) x;`
- 

13. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 16**

1. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata **abstract** o **final**
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
                m();
            }
            catch( Exception s ) {
                System.out.print(2);
                throw( new MyExc1() );
            }
        }
}
```

```
catch( MyExc2 z ) {
    System.out.print(3);
}
finally {
    throw( new MyExc1() );
}
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
}
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

4. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
```

```

public static void main(String[] args) {
    X x = new Z();
    Y y = (Y)x;
    Z z = (Z)y;
    z.f(x); z.f(y); z.f(z);
}
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

5. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

6. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

**9. Qual è l'output di questo codice?**

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**10. Qual è l'output di questo codice?**

```

public interface Building {
    int age;
    public int getAge();
}

```

```

}
class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**11. Qual è l'output di questo codice?**

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**12. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
    }
}

```

```

        catch( MyExc2 j ) {
    }
    catch( Exception v ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
}

A. 13
B. 1
C. 13Exception in thread "main" MyExc2
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;

---

14. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

A. 125
B. 135
C. 1235
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

15. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 
16. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 17

1. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

2. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

3. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

4. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno

- D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

## 9. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {

```

```

            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

## 10. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

## 11. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0

- D. null
  - E. Nessuna delle precedenti
- 

12. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}  
  
A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
```

---

13. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {
```

```
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

15. Qual'è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

16. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- 
- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 18**

1. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. 0

D. null

E. Nessuna delle precedenti

---

2. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

E. Nessuna delle precedenti.

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

A. 12Exception in thread "main" MyExc3

B. 1222222... (ciclo infinito)

C. 12

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
```

```

    }
}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

6. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;

```

```

g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

7. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. 0

D. null

E. Nessuna delle precedenti

---

#### 11. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

#### 12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1

- C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

13. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6
- 

14. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125

- B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

16. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
          Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        }  
    }  
}
```

```
} else {
    System.out.print(0);
}
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
```

A. 110

B. 101

C. 100

D. 011

E. 010

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 19**

1. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
-

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml) {  
        this.ml = ml;  
    }  
    public static void main(String args[]) {  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

- 
5. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

- 
6. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

- 
7. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata **abstract** o **final**
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

- 
8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc3 b ) {  
        }  
        catch( MyExc2 j ) {  
        }  
        catch( Exception v ) {  
            System.out.print(2);  
        }  
        finally {  
            throw( new MyExc2() );  
        }  
    }  
}
```

```

    }
}

static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1

- C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

#### 11. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
E. Nessuna delle precedenti.
- 

#### 12. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
 B. x = y;  
 C. y = m;  
 D. x = m;  
 E. m = (Object []) x;
- 

14. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {

```

```

        House h = new House();
        System.out.println(h.getAge());
    }
}

A. Errore a tempo di compilazione  

B. Errore a tempo di esecuzione  

C. 0  

D. null  

E. Nessuna delle precedenti

```

---

15. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX  

B. XYZ  

C. ZZZ  

D. Errore a tempo di compilazione.  

E. Nessuna delle precedenti.

```

---

16. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {

```

```
}
```

```
static void p() {
```

```
    try {
```

```
        System.out.print(1);
```

```
        throw( new MyExc3() );
```

```
    }
```

```
    catch( MyExc3 y ) {
```

```
        System.out.print(2);
```

```
        throw( new MyExc3() );
```

```
}
```

```
}
```

```
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 20**

1. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
```

```
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
```

```

        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 5. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

#### 6. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

```

```

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

#### 7. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010

---

8. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
}
public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}
```

```
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

13. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {  
            }  
            catch( MyExc2 j ) {  
            }  
            catch( Exception v ) {  
                System.out.print(2);  
            }  
            finally {  
                throw( new MyExc2() );  
            }  
        }  
        static void m()  
            throws Exception {  
            try {  
            }  
            catch( Exception g ) {  
            }  
            finally {  
                System.out.print(3);  
            }  
        }  
}
```

- A. 13
- B. 1

- C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 21**

1. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0

D. null

E. Nessuna delle precedenti

---

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
```

```

        System.out.print(2);
        throw( new MyExc3() );
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse

- B. Una classe può essere dichiarata **abstract** o **final**  
C. Due metodi non possono differire per il tipo di ritorno  
D. Un array non possiede dei membri  
E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

#### 7. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

9. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
        }
```

```
        m();
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc2 j ) {
    }
    catch( Exception v ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

14. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

15. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}
```

```
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 22**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}
```

```
} }
```

```
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}
```

```
class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
}
```

```
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

6. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {
```

```
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

9. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

14. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

```

```

    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {                                     /* QUI */
    String brand ="Nespresso";
    static Integer ml = 20;
}
Coffee() {}
Coffee(int ml){
    this.ml = ml;
}
public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
```

---

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 23**

1. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}

class Mammal extends Animal{
```

```
Mammal() {
    super(6);
    System.out.print(5);
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

3. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX

- B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

4. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

5. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
}
```

```

    }
    public static void main(String args[]) {
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13

- B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
        }
    }
}

```

```
        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 24**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
-

4. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

5. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

6. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null

E. Nessuna delle precedenti

7. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1

- D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

9. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee(){}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
    }  
}
```

```
        return;  
    }  
}  
  
A. 1  
B. 2  
C. 3  
D. 4  
E. 6
```

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

15. Qual è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

16. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
```

```
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

A. 125

- B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 25**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        }
    }
}
```

```
} else {
    System.out.print(0);
}
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125

- B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

5. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

6. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

7. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
            }
```

```

        m();
    }
    catch( Exception s ) {
        System.out.print(2);
        throw( new MyExc1() );
    }
    catch( MyExc2 z ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

11. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {

```

```

}
catch( MyExc2 j ) {
}
catch( Exception v ) {
    System.out.print(2);
}
finally {
    throw( new MyExc2() );
}
}

static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
}
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

### 13. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }

    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 1222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione

### E. Nessuna delle precedenti

---

### 14. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

### 15. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0

- D. null
  - E. Nessuna delle precedenti
- 

16. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 26**

1. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
            }
```

```
    m();
}
catch( Exception s ) {
    System.out.print(2);
    throw( new MyExc1() );
}
catch( MyExc2 z ) {
    System.out.print(3);
}
finally {
    throw( new MyExc1() );
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
                m();
            }
            catch( MyExc3 b ) {
            }
            catch( MyExc2 j ) {
```

```

}
catch( Exception v ) {
    System.out.print(2);
}
finally {
    throw( new MyExc2() );
}
}

static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

5. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
E. Nessuna delle precedenti.
- 

6. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
}

```

```

void p() {
    Boolean b3 = new Boolean(true);
    s7 = s6;
    q(b3, s2, s7, s6);
}
void q(Boolean b2, String s3, String s4,
Object s5) {
    Boolean b1 = new Boolean(true);
    if(b1 == b2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

7. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
-

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

9. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

10. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
```

```
void f(Y y) {
    System.out.print("Y");
}
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

11. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

14. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
}
```

```
Animal(int i) {
    System.out.print(3);
}
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 27

1. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

2. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    /* QUI */
        return;
    }
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

3. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

5. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

7. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

8. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135

C. 1235

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

9. Dire quale delle seguenti affermazioni è vera:

A. Un array ha più di una superclasse

B. Una classe può essere dichiarata **abstract** o **final**

C. Due metodi non possono differire per il tipo di ritorno

D. Un array non possiede dei membri

E. Ogni valore in memoria non è associato ad un tipo di dato particolare

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. Errore a tempo di compilazione

B. 14Exception in thread "main" MyExc1

C. 1

D. 14222222... (ciclo infinito)

E. Nessuna delle precedenti

---

11. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.

D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con **new O.I()**.

E. Nessuna delle precedenti.

---

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

A. 13

- B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;
```

```
public int getAge() {  
    return age;  
}  
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}  
}
```

- 
- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 28**

1. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

3. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `x = (Error []) y;`
- B. `x = y;`
- C. `y = m;`
- D. `x = m;`
- E. `m = (Object []) x;`

---

6. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 z ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc1() );  
            }  
        }  
        static void m()  
            throws Exception {  
            try {  
            }  
            catch( Exception h ) {  
            }  
            finally {  
                System.out.print(4);  
            }  
        }  
    }
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

---

9. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;
```

```

public int getAge() {
    return age;
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
}

```

```

public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

---

12. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

13. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

14. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

15. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
    finally {
        throw( new MyExc2() );
    }
}

```

```
        }
    }
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
```

- ```
        }
    }
A. 13
B. 1
C. 13Exception in thread "main" MyExc2
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 29**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
```

```
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

4. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;

- C. `y = m;`
  - D. `x = m;`
  - E. `m = (Object []) x;`
- 

5. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

7. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
```

```
}
```

```
}
```

```
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
```

```
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
```

```
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
```

```
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

A. 1
B. 2
C. 3
D. 4
E. 6

```

---

12. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 14. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 15. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

#### 16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 30**

1. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
```

```
X x = new Z();
Y y = new Z();
Z z = new Z();
z.f(x); z.f(y); z.f(z);
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
```

```

Mammal(int i) {
    super(6);
    System.out.print(5);
}
}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 5. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}

class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {

```

```

            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
-

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

9. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        }
    }
}
```

```
    } else {
        System.out.print(0);
    }
    if(s3 == s2) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

10. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

11. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

12. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 14Exception in thread "main" MyExc1  
 C. 1  
 D. 14222222... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

### 13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
 B. x = y;  
 C. y = m;  
 D. x = m;  
 E. m = (Object []) x;
- 

### 14. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

### 15. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

### 16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
}

```

```
}

public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}
```

- }
  - A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 31**

1. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

3. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
```

```
private void f(Y y) {
    System.out.print("Y");
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

4. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

5. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

6. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {  
            }  
        }  
}
```

```
        catch( MyExc2 j ) {  
        }  
        catch( Exception v ) {  
            System.out.print(2);  
        }  
        finally {  
            throw( new MyExc2() );  
        }  
    }  
    static void m()  
        throws Exception {  
        try {  
        }  
        catch( Exception g ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        } catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        } catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
 B. 1222222... (ciclo infinito)  
 C. 12  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
    throws Exception {
    try {
    }
    catch( Exception h ) {
    }
}

```

```

        finally {
            System.out.print(4);
        }
    }

```

- A. Errore a tempo di compilazione  
 B. 14Exception in thread "main" MyExc1  
 C. 1  
 D. 14222222... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

#### 11. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
 B. 101  
 C. 100  
 D. 011  
 E. 010
- 

#### 12. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

14. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

16. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

```

```
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 32

1. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc3 b ) {  
        }  
        catch( MyExc2 j ) {  
        }  
        catch( Exception v ) {  
            System.out.print(2);  
        }  
    finally {  
        throw( new MyExc2() );  
    }  
    static void m()
```

```

throws Exception {
    try {
    }
    catch( Exception g ) {
    }
    finally {
        System.out.print(3);
    }
}

```

- A. 13  
 B. 1  
 C. 13Exception in thread "main" MyExc2  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

5. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
 B. Una classe può essere dichiarata **abstract** o **final**  
 C. Due metodi non possono differire per il tipo di ritorno  
 D. Un array non possiede dei membri  
 E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

6. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
 B. XYZ  
 C. ZZZ  
 D. Errore a tempo di compilazione.  
 E. Nessuna delle precedenti.
- 

7. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

9. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
          Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

```
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

11. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

12. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- ```
}
```
- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

16. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}

class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
```

```
}

}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 33**

**1. Qual è l'output di questo codice?**

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**2. Qual è l'output di questo codice?**

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
```

```
    }
    catch( Exception v ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**3. Dato un file contenente il seguente codice:**

```
package pk;
public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

---

4. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ

C. ZZZ

- D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

7. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
    }  
}
```

```

        c1 = null;
        c3 = c1;
    /* QUI */
    return;
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6
- 

#### 9. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 10. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;

E. m = (Object []) x;

---

#### 11. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;  
B. g = (Boolean) s;  
C. b = (Integer) g;  
D. g = (Boolean) b;  
E. Nessuno dei precedenti
- 

#### 12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011

## 13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

## 14. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

```

```

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

## 15. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

## 16. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}

class Mammal extends Animal{
    Mammal(int i) {

```

```
super(6);
System.out.print(5);
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

---

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti



prof. Piero A. Bonatti

26 giugno 2019

Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 34

1. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

3. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
    }  
}
```

```
    Mammal m = new Mammal();  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

4. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

5. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 7. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 12222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {}  
            catch( MyExc2 j ) {}  
            catch( Exception v ) {
```

```
                System.out.print(2);  
            }  
            finally {  
                throw( new MyExc2() );  
            }  
        }  
        static void m()  
            throws Exception {  
                try {  
                }  
                catch( Exception g ) {  
                }  
                finally {  
                    System.out.print(3);  
                }  
            }  
        }  
    }
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal {  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

13. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 101
- C. 100
- D. 011

E. 010

14. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

16. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 35**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
```

```
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
- B. 2

C. 3

D. 4

E. 6

---

4. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

5. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

A. b = (Integer) s;

B. g = (Boolean) s;

C. b = (Integer) g;

D. g = (Boolean) b;

E. Nessuno dei precedenti

---

6. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 z ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc1() );  
            }  
        }  
        static void m()  
            throws Exception {  
            try {  
            }  
            catch( Exception h ) {  
            }  
            finally {  
                System.out.print(4);  
            }  
        }  
}
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

11. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

13. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

---

14. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

---

15. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
}
```

```
}

public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

A. Errore a tempo di compilazione
```

- B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 36**

1. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

2. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
```

```

    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new O.I()`.
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {

```

```

public static void main(String [] argv)
    throws Exception {
    try {
        System.out.print(1);
        m();
    }
    catch( Exception s ) {
        System.out.print(2);
        throw( new MyExc1() );
    }
    catch( MyExc2 z ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

12. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
}

```

```

    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

```

```
class Z extends Y {                                }
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
```

- }
- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 37

1. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

2. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

3. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

4. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 14Exception in thread "main" MyExc1  
 C. 1  
 D. 14222222... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {

```

```

            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3  
 B. 1222222... (ciclo infinito)  
 C. 12  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 7. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {

```

```

        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13  
 B. 1  
 C. 13Exception in thread "main" MyExc2  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

9. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1  
 B. 2  
 C. 3  
 D. 4  
 E. 6
- 

```

        Integer b;
        Boolean g;
        Object s;
        g = new Boolean(true);
        b = new Integer(50);
        s = new Integer(50);
    
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;  
 B. g = (Boolean) s;  
 C. b = (Integer) g;  
 D. g = (Boolean) b;  
 E. Nessuno dei precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
 B. 101  
 C. 100  
 D. 011  
 E. 010
- 

10. Date le dichiarazioni:

12. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
 B. XYZ  
 C. ZZZ  
 D. Errore a tempo di compilazione.  
 E. Nessuna delle precedenti.
- 

13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
 B. x = y;  
 C. y = m;  
 D. x = m;  
 E. m = (Object []) x;
- 

14. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{

```

```

String type = "Residential";
int age;
public int getAge() {
    return age;
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

15. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
 B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
 C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
 D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
 E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 38

1. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

2. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

3. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

4. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}
```

```

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 5. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 6. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

#### 7. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    }
}

```

```

        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }

```

```

        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 12222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

11. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
B. Una classe può essere dichiarata abstract o final  
C. Due metodi non possono differire per il tipo di ritorno  
D. Un array non possiede dei membri  
E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {

```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 110  
 B. 101  
 C. 100  
 D. 011  
 E. 010
- 

13. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.  
 B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.  
 C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.  
 D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().  
 E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
    finally {

```

```

        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}

```

- A. 13  
 B. 1  
 C. 13Exception in thread "main" MyExc2  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();

```

```
Coffee c2 = new Coffee();  
Coffee c3 = new Coffee(40);  
c1 = null;  
c3 = c1;  
/* QUI */  
return;  
}  
}
```

---

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 39**

1. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3

- B. 12222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata `abstract` o `final`
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

7. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}
```

```
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( MyExc3 b ) {  
        }  
        catch( MyExc2 j ) {  
        }  
        catch( Exception v ) {  
            System.out.print(2);  
        }  
        finally {  
            throw( new MyExc2() );  
        }  
    }  
    static void m()  
        throws Exception {  
        try {  
        }  
        catch( Exception g ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
}
```

- A. 13
- B. 1

C. 13Exception in thread "main" MyExc2

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

9. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 110

B. 101

C. 100

D. 011

E. 010

---

10. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

A. x = (Error []) y;

B. x = y;

C. y = m;

D. x = m;

E. m = (Object []) x;

---

11. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

A. 125

B. 135

C. 1235

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

12. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
    }
}
```

```

    Z z = (Z)y;
    z.f(x); z.f(y); z.f(z);
}
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

13. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

14. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}

```

```

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

16. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 40**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
```

```
}  
catch( MyExc3 y ) {  
    System.out.print(2);  
    throw( new MyExc3() );  
}  
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

4. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0

- D. null
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
}
```

```
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {

```

```

private void f(Y y) {
    System.out.print("Y");
}
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.

```

---

11. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.

```

E. Nessuna delle precedenti.

---

12. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

13. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

14. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;
```

```
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}  
  
A. 110  
B. 101  
C. 100  
D. 011  
E. 010
```

---

15. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

16. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal {  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}
```

```
}

}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 41**

1. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

2. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

3. Qual è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}
```

```
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

4. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
}  
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

- C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

6. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

7. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

9. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
- B. 135

C. 1235

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
```

```

        System.out.print(2);
        throw( new MyExc1() );
    }
    catch( MyExc2 z ) {
        System.out.print(3);
    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
    try {
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
B. Una classe può essere dichiarata abstract o final  
C. Due metodi non possono differire per il tipo di ritorno  
D. Un array non possiede dei membri  
E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);

```

```

        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

15. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 42

1. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata **abstract** o **final**
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand = "Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml) {  
        this.ml = ml;  
    }  
    public static void main(String args[]) {  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

3. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}
```

```
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

4. Qual è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX

- B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

6. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
    }
}
```

```
    Mammal m = new Mammal();
}
}
A. 125
B. 135
C. 1235
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```

---

7. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
    }
}
```

```

        catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 9. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
    }
}

```

```

        catch( MyExc3 b ) {
    }
    catch( MyExc2 j ) {
    }
    catch( Exception v ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 11. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```
    }
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

---

12. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

13. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

---

14. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

15. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 43

1. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

2. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

3. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.

- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

- E. Nessuna delle precedenti.

4. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

5. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

6. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends Exception { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {  
            }  
            catch( MyExc2 j ) {  
            }  
            catch( Exception v ) {  
                System.out.print(2);  
            }  
            finally {  
                throw( new MyExc2() );  
            }  
    }  
    static void m()  
        throws Exception {  
            try {  
            }  
            catch( Exception h ) {  
            }  
            finally {  
                System.out.print(4);  
            }  
        }  
}
```

```
        catch( Exception g ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
            catch( MyExc2 z ) {  
                System.out.print(3);  
            }  
            finally {  
                throw( new MyExc1() );  
            }  
        }  
        static void m()  
            throws Exception {  
                try {  
                }  
                catch( Exception h ) {  
                }  
                finally {  
                    System.out.print(4);  
                }  
            }  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

13. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

14. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal {  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
    }  
}
```

```
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

16. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building {  
    String type = "Residential";
```

```
int age;
public int getAge() {
    return age;
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

**Prova n. 44**

1. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

2. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
    }
    if(s4 == s5) {
        System.out.print(1);
    } else {
        System.out.print(0);
    }
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

3. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. Errore a tempo di compilazione

B. 14Exception in thread "main" MyExc1

C. 1

D. 14222222... (ciclo infinito)

E. Nessuna delle precedenti

---

5. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
}
```

```
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

6. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

A. 125

B. 135

C. 1235

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. `x = (Error []) y;`
  - B. `x = y;`
  - C. `y = m;`
  - D. `x = m;`
  - E. `m = (Object []) x;`
- 

8. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (Integer) s;`
  - B. `g = (Boolean) s;`
  - C. `b = (Integer) g;`
  - D. `g = (Boolean) b;`
  - E. Nessuno dei precedenti
- 

11. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

**13. Qual è l'output di questo codice?**

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**14. Qual è l'output di questo codice?**

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**15. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Dire quale delle seguenti affermazioni è vera:

---

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata **abstract** o **final**
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 45**

1. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
```

```
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
```

```

        System.out.print(1);
    } else {
        System.out.print(0);
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

**4. Qual è l'output di questo codice?**

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**5. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?**

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    }
}

```

```

/* QUI */
return;
}
}
A. 1
B. 2
C. 3
D. 4
E. 6

```

---

**6. Qual è l'output di questo programma?**

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**7. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    }
}

```

```

    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 8. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 9. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

#### 10. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

#### 11. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

12. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {}  
            catch( MyExc2 j ) {}  
            catch( Exception v ) {  
                System.out.print(2);  
            }  
            finally {  
                throw( new MyExc2() );  
            }  
        }  
        static void m()  
    }
```

```
    throws Exception {  
        try {  
        }  
        catch( Exception g ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }  
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

15. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con `new O.I()`.
  - E. Nessuna delle precedenti.
- 

16. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata `abstract` o `final`
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 46**

1. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    finally {

```

```

        throw( new MyExc1() );
    }
}

static void m()
throws Exception {
    try {
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
B. 12222222... (ciclo infinito)  
C. 12  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

#### 7. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```
class Test {
```

#### 6. Quale output si ottiene invocando il metodo p?

```

public static void main(String[] args) {
    X x = new Z();
    Y y = (Y)x;
    Z z = (Z)y;
    z.f(x); z.f(y); z.f(z);
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

8. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 
13. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6
- 

14. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

```

```

    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

15. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
}
```

```
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 47**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

4. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

5. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

6. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

7. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

8. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

12. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
        }  
}
```

```

        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 
13. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1  
B. 2  
C. 3  
D. 4  
E. 6
- 
14. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti

---

15. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

A. XXX  
B. XYZ  
C. ZZZ

- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata `abstract` o `final`

- C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 48**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
```

```
}  
catch( MyExc3 y ) {  
    System.out.print(2);  
    throw( new MyExc3() );  
}  
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

5. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
```

```
throws Exception {
    try {
        System.out.print(1);
        m();
    }
    catch( MyExc3 b ) {
    }
    catch( MyExc2 j ) {
    }
    catch( Exception v ) {
        System.out.print(2);
    }
    finally {
        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
    try {
    }
    catch( Exception g ) {
    }
    finally {
        System.out.print(3);
    }
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

8. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

9. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
```

```
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
```

```

Boolean b1 = new Boolean(true);
if(b1 == b2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

13. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

14. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

```

```

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

15. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 49**

1. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
} }
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Qual'è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
}

public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null

E. Nessuna delle precedenti

---

4. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
          Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

5. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

6. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

7. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
```

```

}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Quanti oggetti *al massimo* possono essere deallocated mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2
- C. 3

D. 4

E. 6

---

15. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

A. 125
B. 135
C. 1235
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {

```

```
}

catch( Exception h ) {
}

finally {
    System.out.print(4);
}

}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 50**

1. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

A. 1
B. 2
C. 3
D. 4
E. 6
```

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

```

    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

6. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

class Test {
    public static void main(String[] args){
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 9. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);

```

```

                m();
            }
            catch( Exception s ) {
                System.out.print(2);
                throw( new MyExc1() );
            }
            catch( MyExc2 z ) {
                System.out.print(3);
            }
            finally {
                throw( new MyExc1() );
            }
        }
        static void m()
            throws Exception {
            try {
            }
            catch( Exception h ) {
            }
            finally {
                System.out.print(4);
            }
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 11. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

#### 12. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

**13. Qual'è l'output di questo programma?**

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

**14. Qual è l'output di questo codice?**

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";

```

```

    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**15. Qual è l'output di questo codice?**

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

**16. Quale output si ottiene invocando il metodo p?**

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        }
    }
}

```

```
} else {
    System.out.print(0);
}
if(s3 == s2) {
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
```

A. 110

B. 101

C. 100

D. 011

E. 010

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 51**

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

```
System.out.print(1);
Mammal m = new Mammal();
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }

public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {

```

```

        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

6. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }

public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {

```

```

        System.out.print(2);
        throw( new MyExc3() );
    }
}

```

- A. 12Exception in thread "main" MyExc3  
 B. 1222222... (ciclo infinito)  
 C. 12  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
 B. x = y;  
 C. y = m;  
 D. x = m;  
 E. m = (Object []) x;
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
 B. Una classe può essere dichiarata abstract o final  
 C. Due metodi non possono differire per il tipo di ritorno  
 D. Un array non possiede dei membri  
 E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
                m();
            }
            catch( MyExc3 b ) {
            }
            catch( MyExc2 j ) {
            }
        }
}

```

```

        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
            try {
            }
            catch( Exception g ) {
            }
            finally {
                System.out.print(3);
            }
        }
}

```

- A. 13  
 B. 1  
 C. 13Exception in thread "main" MyExc2  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

**13. Date le dichiarazioni:**

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

**14. Qual'è l'output di questo programma?**

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

**15. Qual è l'output di questo codice?**

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

16. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
}
```

```
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 52**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
```

```
static Integer ml = 20;

Coffee() {}
Coffee(int ml) {
    this.ml = ml;
}
public static void main(String args[]) {
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}

A. 1
B. 2
C. 3
D. 4
E. 6
```

---

3. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
}
```

```

static void m()
    throws Exception {
    try {
    }
    catch( Exception g ) {
    }
    finally {
        System.out.print(3);
    }
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

```

    }
}

class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 4. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 6. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010

#### 5. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}

```

---

7. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

8. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

9. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione

C. 0

D. null

E. Nessuna delle precedenti

---

10. Qual è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

11. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

12. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
        }
    }
}
```

```
        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}
}
```

---

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 53**

1. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    finally {
        throw( new MyExc1() );
    }
}
```

```
}
static void m()
throws Exception {
try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

4. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
```

```

int age;
public int getAge() {
    return age;
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {

```

```

            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]) {
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}
```

```
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}
```

```
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

9. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}
```

```
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}
```

```
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
    }  
}
```

```
        z.f(x); z.f(y); z.f(z);  
    }  
}  
  
A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
```

---

10. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

11. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
    }  
}
```

```

        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
 B. 1222222... (ciclo infinito)  
 C. 12  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

13. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
 B. x = y;  
 C. y = m;  
 D. x = m;  
 E. m = (Object []) x;
- 

14. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

16. Dire quale delle seguenti affermazioni è vera:

---

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata **abstract** o **final**
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 54**

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand = "Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml) {
        this.ml = ml;
    }
    public static void main(String args[]) {
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
```

```
        System.out.print(3);
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 5. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

#### 6. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 7. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 8. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null

E. Nessuna delle precedenti

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

A. Errore a tempo di compilazione

B. 14Exception in thread "main" MyExc1

C. 1

D. 14222222... (ciclo infinito)

E. Nessuna delle precedenti

10. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.

C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.

D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

E. Nessuna delle precedenti.

11. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

A. 110

B. 101

C. 100

D. 011

E. 010

12. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. `b = (Integer) s;`
- B. `g = (Boolean) s;`
- C. `b = (Integer) g;`
- D. `g = (Boolean) b;`
- E. Nessuno dei precedenti

---

13. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

14. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
```

```
        System.out.println(h.getAge());
    }
}

A. Errore a tempo di compilazione
B. Errore a tempo di esecuzione
C. 0
D. null
E. Nessuna delle precedenti
```

---

15. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

16. Qual è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 55**

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

2. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

3. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

4. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

5. Qual è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

7. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

10. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);

```

```

            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13  
 B. 1  
 C. 13Exception in thread "main" MyExc2  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 14. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
    }
}

```

```

        Mammal m = new Mammal();
    }
}
A. 125
B. 135
C. 1235
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

#### 15. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
A. 125
B. 135
C. 1235
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

#### 16. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {

```

```
    System.out.print(0);                                }
}
if(s3 == s2) {
    System.out.print(1);                                A. 110
} else {
    System.out.print(0);                                B. 101
}
if(s4 == s5) {
    System.out.print(1);                                C. 100
} else {
    System.out.print(0);                                D. 011
}
System.out.print(0);                                E. 010
}
```



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 56**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
```

```
static Integer ml = 20;

Coffee() {}
Coffee(int ml) {
    this.ml = ml;
}
public static void main(String args[]) {
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
    /* QUI */
    return;
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null

E. Nessuna delle precedenti

4. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.

5. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```

        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare

7. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception q ) {

```

```

    }
    finally {
        System.out.print(3);
    }
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

8. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;
- 

9. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione
- 

E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
-

12. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

13. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

14. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

15. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

16. Qual è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}
```

```
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- ```
    }  
  
A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
```
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 57**

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

2. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

5. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

6. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

---

E. Nessuna delle precedenti.

---

7. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( Exception s ) {  
            System.out.print(2);  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 z ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc1() );  
        }  
    }  
    static void m()  
        throws Exception {
```

```

try {
}
catch( Exception h ) {
}
finally {
    System.out.print(4);
}
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

9. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
}

```

```

        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

11. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

12. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {
```

```
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

15. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal {  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-

16. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono

corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 58**

1. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
-

4. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {
```

```
public static void main(String [] argv)  
throws Exception {  
    try {  
        System.out.print(1);  
        m();  
    }  
    catch( Exception s ) {  
        System.out.print(2);  
        throw( new MyExc1() );  
    }  
    catch( MyExc2 z ) {  
        System.out.print(3);  
    }  
    finally {  
        throw( new MyExc1() );  
    }  
}  
static void m()  
throws Exception {  
    try {  
    }  
    catch( Exception h ) {  
    }  
    finally {  
        System.out.print(4);  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

8. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;

D. `x = m;`  
E. `m = (Object []) x;`

---

9. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

10. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
```

```
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

11. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

12. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

14. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 59**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
} }
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

---

3. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione

E. Nessuna delle precedenti

4. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

5. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

6. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
}
```

```
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

7. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata **abstract** o **final**
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

8. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

9. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

#### 10. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

- C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 11. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

#### 12. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- B. 101
- C. 100
- D. 011
- E. 010

---

13. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12

- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
- B. 1
- C. 13Exception in thread "main" MyExc2
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

16. Qual è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {                                }
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
```

- }
- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 60**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
}
```

```
}  
Animal(int i) {  
    System.out.print(3);  
}  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }
}  
A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
```

---

3. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }
}  
public static void main(String args[]){  
    Coffee c1 = new Coffee();  
    Coffee c2 = new Coffee();  
    Coffee c3 = new Coffee(40);  
    c1 = null;  
    c3 = c1;  
    /* QUI */  
    return;  
}
```

- A. 1
- B. 2

C. 3

D. 4

E. 6

---

4. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

A. 12Exception in thread "main" MyExc3

B. 1222222... (ciclo infinito)

C. 12

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

5. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
```

```
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

6. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. 0

D. null

E. Nessuna delle precedenti

---

7. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

A. b = (Integer) s;

B. g = (Boolean) s;

C. b = (Integer) g;

D. g = (Boolean) b;

E. Nessuno dei precedenti

---

8. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

10. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

11. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

12. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;

E. `m = (Object []) x;`

---

13. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
          Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

16. Qual'è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}
```

```
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 61**

1. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione

- B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
```

```

public static void main(String args[]) {
    System.out.print(1);
    Mammal m = new Mammal();
}

```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    }
}

```

```

        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

7. Qual è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.
- 

8. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

9. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

10. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }
```

```
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

11. Quale output si ottiene invocando il metodo p?

```
class F {  
    private String s1 = "";  
    private String s2 = s1.concat("a");  
    private String s6 = new String("abcd");  
    private String s7;  
    void p() {  
        Boolean b3 = new Boolean(true);  
        s7 = s6;  
        q(b3, s2, s7, s6);  
    }  
    void q(Boolean b2, String s3, String s4,  
           Object s5) {  
        Boolean b1 = new Boolean(true);  
        if(b1 == b2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s3 == s2) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
        if(s4 == s5) {  
            System.out.print(1);  
        } else {  
            System.out.print(0);  
        }  
    }  
}
```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

---

12. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        } catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        } catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;

- D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

15. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

16. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
    }
}

```

```
        }
    finally {
        throw( new MyExc2() );
    }
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
        }
    }
}
A. 13
B. 1
C. 13Exception in thread "main" MyExc2
D. Errore a tempo di compilazione
E. Nessuna delle precedenti
```

---



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 62**

1. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
```

```
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
```

```

    }
}

class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
- B. 1

C. 13Exception in thread "main" MyExc2

- D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 7. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}

class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}

class Rabbit extends Mammal {

```

```

public static void main(String args[]) {
    System.out.print(1);
    Mammal m = new Mammal();
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse

- B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
        static void p() {
            try {
                System.out.print(1);
                throw( new MyExc3() );
            }
            catch( MyExc3 y ) {
                System.out.print(2);
                throw( new MyExc3() );
            }
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3
  - B. 12222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    }
}

```

```

    /* QUI */
    return;
}
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

13. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

14. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

15. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

16. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {

```

```
    System.out.print(1);
} else {
    System.out.print(0);
}
if(s4 == s5) {
    System.out.print(1);
} else {
    System.out.print(0);
}
}
```

- 
- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 63

1. Qual è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
}
```

```
}  
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

4. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

5. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 6. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 7. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

#### 8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

#### 9. Qual è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

10. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }

    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
- B. 2

- C. 3
- D. 4
- E. 6

11. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

A. 125
B. 135
C. 1235
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

12. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

13. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 14. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 14Exception in thread "main" MyExc1  
 C. 1  
 D. 14222222... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

#### 15. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}

```

- A. 12Exception in thread "main" MyExc3  
 B. 12222222... (ciclo infinito)  
 C. 12  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

#### 16. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {

```

```
System.out.print(1);  
} else {  
    System.out.print(0);  
}  
  
if(s3 == s2) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}  
  
if(s4 == s5) {  
    System.out.print(1);  
} else {  
    System.out.print(0);  
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 64**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
```

```
public static void main(String args[]) {
    System.out.print(1);
    Mammal m = new Mammal();
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13  
B. 1  
C. 13Exception in thread "main" MyExc2  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

#### 5. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. 0

D. null

E. Nessuna delle precedenti

---

#### 6. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
B. Una classe può essere dichiarata abstract o final  
C. Due metodi non possono differire per il tipo di ritorno  
D. Un array non possiede dei membri  
E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

#### 7. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

#### 8. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

9. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

10. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
C. ZZZ
D. Errore a tempo di compilazione.
E. Nessuna delle precedenti.
```

---

11. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

A. XXX
B. XYZ
```

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

12. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

---

13. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

14. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( Exception s ) {  
                System.out.print(2);  
                throw( new MyExc1() );  
            }  
        }  
}
```

```
        catch( MyExc2 z ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc1() );  
        }  
    }  
    static void m()  
        throws Exception {  
            try {  
            }  
            catch( Exception h ) {  
            }  
            finally {  
                System.out.print(4);  
            }  
        }  
}
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

---

15. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

16. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {                                     /* QUI */
    String brand ="Nespresso";
    static Integer ml = 20;
}
Coffee() {}
Coffee(int ml){
    this.ml = ml;
}
public static void main(String args[]){
    Coffee c1 = new Coffee();
    Coffee c2 = new Coffee();
    Coffee c3 = new Coffee(40);
    c1 = null;
    c3 = c1;
```

---

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

Prova n. 65

1. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
- B. XYZ
- C. ZZZ
- D. Errore a tempo di compilazione.
- E. Nessuna delle precedenti.

2. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
}
```

```
}  
public static void main(String args[]) {  
    House h = new House();  
    System.out.println(h.getAge());  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

3. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

4. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}
```

```

}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

6. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

```

    }
}

A. 13
B. 1
C. 13Exception in thread "main" MyExc2
D. Errore a tempo di compilazione
E. Nessuna delle precedenti

```

---

8. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

10. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

11. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

12. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

**14. Qual è l'output di questo codice?**

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {

```

```

        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

**15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?**

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

**16. Qual è l'output di questo codice?**

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
}

```

```
static void p() {  
    try {  
        System.out.print(1);  
        throw( new MyExc3() );  
    }  
    catch( MyExc3 y ) {  
        System.out.print(2);  
        throw( new MyExc3() );  
    }  
}
```

```
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 66**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
            try {
                System.out.print(1);
                m();
            }
            catch( MyExc3 b ) {
            }
            catch( MyExc2 j ) {
            }
            catch( Exception v ) {
```

```
                System.out.print(2);
            }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
            try {
            }
            catch( Exception g ) {
            }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[])
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null

E. Nessuna delle precedenti

---

4. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

5. Qual è l'output di questo codice?

```
class MyExc1 extends Error {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
    }
```

```
        System.out.print(2);  
        throw( new MyExc3() );  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
- B. 1222222... (ciclo infinito)
- C. 12
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

---

6. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends MyExc1 {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
        try {  
            System.out.print(1);  
            m();  
        }  
        catch( Exception s ) {  
            System.out.print(2);  
            throw( new MyExc1() );  
        }  
        catch( MyExc2 z ) {  
            System.out.print(3);  
        }  
        finally {  
            throw( new MyExc1() );  
        }  
    }  
    static void m()  
        throws Exception {  
        try {  
        }  
        catch( Exception h ) {  
        }  
        finally {  
            System.out.print(4);  
        }  
    }  
}
```

- A. Errore a tempo di compilazione
- B. 14Exception in thread "main" MyExc1
- C. 1
- D. 14222222... (ciclo infinito)
- E. Nessuna delle precedenti

---

7. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

#### 8. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

#### 9. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

#### 10. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

#### 11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
    }
}

```

```

        c1 = null;
        c3 = c1;
    /* QUI */
    return;
}
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata abstract o final
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

14. Date le dichiarazioni:

```

Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
  - B. 101
  - C. 100
  - D. 011
  - E. 010
- 

16. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.

- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 67**

1. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
```

```
public static void main(String args[]) {
    System.out.print(1);
    Mammal m = new Mammal();
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Date le dichiarazioni:

```
Object [] m;
Error [] x;
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
  - B. x = y;
  - C. y = m;
  - D. x = m;
  - E. m = (Object []) x;
- 

4. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
    }
}
```

```

        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)  
E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

6. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {

```

```

        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101  
C. 100  
D. 011  
E. 010
- 

7. Qual è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

8. Qual è l'output di questo codice?

```
public interface Building {  
    int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

10. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
            throw( new MyExc3() );  
        }  
        catch( MyExc3 y ) {  
            System.out.print(2);  
            throw( new MyExc3() );  
        }  
    }  
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Qual'è l'output di questo programma?

```
class X {  
    private void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

12. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception {}  
class MyExc2 extends Exception {}  
class MyExc3 extends MyExc2 {}  
public class B1 {  
    public static void main(String [] argv)  
        throws Exception {  
            try {  
                System.out.print(1);  
                m();  
            }  
            catch( MyExc3 b ) {}  
            catch( MyExc2 j ) {}  
            catch( Exception v ) {  
                System.out.print(2);  
            }  
            finally {  
                throw( new MyExc2() );  
            }  
        }  
        static void m()  
            throws Exception {
```

```
        try {  
        }  
        catch( Exception g ) {  
        }  
        finally {  
            System.out.print(3);  
        }  
    }
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Dato un file contenente il seguente codice:

```
package pk;  
  
public class O {  
    static class I {  
    }  
    //qui sono i metodi di O...  
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;
```

```
/* QUI */  
    return;  
}  
}  
  
A. 1  
B. 2  
C. 3  
D. 4  
E. 6
```

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 68**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}
```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}
```

```
} }
```

```
class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}
```

```
class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
}
```

```
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}

```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

5. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

```

```

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

6. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110
- B. 101
- C. 100
- D. 011
- E. 010

---

7. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;
- D. g = (Boolean) b;
- E. Nessuno dei precedenti

---

8. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

---

9. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata abstract o final

- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

---

10. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
- E. Nessuna delle precedenti.

---

11. Qual è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

12. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
- B. Errore a tempo di esecuzione
- C. 0
- D. null
- E. Nessuna delle precedenti

13. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;
- E. m = (Object []) x;

14. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}
```

```
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125
- B. 135
- C. 1235
- D. Errore a tempo di compilazione
- E. Nessuna delle precedenti

15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}
```

- A. 1
- B. 2
- C. 3
- D. 4
- E. 6

16. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
    static void p() {  
        try {  
            System.out.print(1);  
        }
```

```
        throw( new MyExc3() );
    }
    catch( MyExc3 y ) {
        System.out.print(2);
        throw( new MyExc3() );
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 69**

1. Qual'è l'output di questo programma?

```
class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

2. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

3. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

4. Qual'è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
```

```

        System.out.print(3);
    }
    finally {
        throw( new MyExc1() );
    }
}
static void m()
throws Exception {
    try {
    }
    catch( Exception h ) {
    }
    finally {
        System.out.print(4);
    }
}
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

6. Qual è l'output di questo codice?

```

class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
```

```

        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 12222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

7. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee() {}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}
```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

8. Qual'è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

9. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

```

        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

A. 110
B. 101
C. 100
D. 011
E. 010

```

---

10. Qual'è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

11. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;
- B. x = y;
- C. y = m;
- D. x = m;

E. `m = (Object []) x;`

---

12. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata `abstract` o `final`
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

13. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}
```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 70**

1. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

2. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
```

```
} catch( MyExc3 b ) {
} catch( MyExc2 j ) {
} catch( Exception v ) {
    System.out.print(2);
}
finally {
    throw( new MyExc2() );
}
}
static void m()
throws Exception {
try {
}
catch( Exception g ) {
}
finally {
    System.out.print(3);
}
}
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.

D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

E. Nessuna delle precedenti.

---

4. Qual è l'output di questo programma?

```
class X {  
    void f(X x) {  
        System.out.print("X");  
    }  
}  
  
class Y extends X {  
    private void f(Y y) {  
        System.out.print("Y");  
    }  
}  
  
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args){  
        X x = new Z();  
        Y y = (Y)x;  
        Z z = (Z)y;  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

A. XXX

B. XYZ

C. ZZZ

D. Errore a tempo di compilazione.

E. Nessuna delle precedenti.

---

5. Date le dichiarazioni:

```
Integer b;  
Boolean g;  
Object s;  
g = new Boolean(true);  
b = new Integer(50);  
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

A. b = (Integer) s;

B. g = (Boolean) s;

C. b = (Integer) g;

D. g = (Boolean) b;

E. Nessuno dei precedenti

---

6. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. 0

D. null

E. Nessuna delle precedenti

---

7. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge() {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

A. Errore a tempo di compilazione

B. Errore a tempo di esecuzione

C. 0

D. null

E. Nessuna delle precedenti

---

8. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null  
E. Nessuna delle precedenti
- 

#### 9. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}

```

- A. 110  
B. 101

C. 100

D. 011

E. 010

---

#### 10. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;
- 

#### 11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
B. 14Exception in thread "main" MyExc1  
C. 1  
D. 14222222... (ciclo infinito)

E. Nessuna delle precedenti

---

12. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

A. 125

B. 135

C. 1235

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

13. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
- B. Una classe può essere dichiarata `abstract` o `final`
- C. Due metodi non possono differire per il tipo di ritorno
- D. Un array non possiede dei membri
- E. Ogni valore in memoria non è associato ad un tipo di dato particolare

---

14. Qual'è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
    }  
}
```

```
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

A. 125

B. 135

C. 1235

D. Errore a tempo di compilazione

E. Nessuna delle precedenti

---

15. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```
public class Coffee {  
    String brand ="Nespresso";  
    static Integer ml = 20;  
  
    Coffee() {}  
    Coffee(int ml){  
        this.ml = ml;  
    }  
    public static void main(String args[]){  
        Coffee c1 = new Coffee();  
        Coffee c2 = new Coffee();  
        Coffee c3 = new Coffee(40);  
        c1 = null;  
        c3 = c1;  
        /* QUI */  
        return;  
    }  
}  
  
A. 1  
B. 2  
C. 3  
D. 4  
E. 6
```

---

16. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }  
class MyExc2 extends MyExc1 { }  
class MyExc3 extends MyExc2 { }  
public class C1 {  
    public static void main(String [] argv) {  
        try {  
            p();  
        }  
        catch( MyExc2 r ) {  
        }  
    }  
}
```

```
static void p() {  
    try {  
        System.out.print(1);  
        throw( new MyExc3() );  
    }  
    catch( MyExc3 y ) {  
        System.out.print(2);  
        throw( new MyExc3() );  
    }  
}
```

```
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 71**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {
        }
        finally {
            System.out.print(3);
        }
    }
}
```

- A. 13
  - B. 1
  - C. 13Exception in thread "main" MyExc2
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Qual'è l'output di questo programma?

```
class X {
    private void f(X x) {
        System.out.print("X");
    }
}
```

```
} }
```

```
class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}
```

```
class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}
```

```
class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = new Z();
        Z z = new Z();
        z.f(x); z.f(y); z.f(z);
    }
}
```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

3. Dato un file contenente il seguente codice:

```
package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}
```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
- B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
- C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.

- D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().

- E. Nessuna delle precedenti.

---

4. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal(int i) {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```
public interface Building {  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;  
    public int getAge(int age) {  
        return age;  
    }  
    public static void main(String args[]) {  
        House h = new House();  
        System.out.println(h.getAge());  
    }  
}
```

- A. Errore a tempo di compilazione  
B. Errore a tempo di esecuzione  
C. 0  
D. null

- E. Nessuna delle precedenti
- 

6. Qual è l'output di questo programma?

```
class Animal {  
    Animal() {  
        System.out.print(2);  
    }  
    Animal(int i) {  
        System.out.print(3);  
    }  
}  
class Mammal extends Animal{  
    Mammal() {  
        super(6);  
        System.out.print(5);  
    }  
}  
class Rabbit extends Mammal {  
    public static void main(String args[]) {  
        System.out.print(1);  
        Mammal m = new Mammal();  
    }  
}
```

- A. 125  
B. 135  
C. 1235  
D. Errore a tempo di compilazione  
E. Nessuna delle precedenti
- 

7. Date le dichiarazioni:

```
Object [] m;  
Error [] x;  
Boolean [] y;
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
B. x = y;  
C. y = m;  
D. x = m;  
E. m = (Object []) x;
- 

8. Qual è l'output di questo codice?

```
public interface Building {  
    int age;  
    public int getAge();  
}  
  
class House implements Building{  
    String type = "Residential";  
    int age;
```

```

public int getAge() {
    return age;
}
public static void main(String args[]) {
    House h = new House();
    System.out.println(h.getAge());
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

9. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX
  - B. XYZ
  - C. ZZZ
  - D. Errore a tempo di compilazione.
  - E. Nessuna delle precedenti.
- 

10. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }

```

```

public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione
  - B. 14Exception in thread "main" MyExc1
  - C. 1
  - D. 14222222... (ciclo infinito)
  - E. Nessuna delle precedenti
- 

11. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
        /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

12. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
  - B. g = (Boolean) s;
  - C. b = (Integer) g;
  - D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

13. Qual è l'output di questo codice?

```
public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

14. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

15. Quale output si ottiene invocando il metodo p?

```
class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
    }
}
```

```
    }  
}  
  
A. 110  
B. 101  
C. 100  
D. 011  
E. 010
```

---

16. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse
  - B. Una classe può essere dichiarata **abstract** o **final**
  - C. Due metodi non possono differire per il tipo di ritorno
  - D. Un array non possiede dei membri
  - E. Ogni valore in memoria non è associato ad un tipo di dato particolare
-



Segnare nell'ultimo foglio le risposte alle seguenti domande a scelta multipla tenendo presente che ogni domanda richiede una sola risposta.

---

**Prova n. 72**

1. Qual è l'output di questo codice?

```
class MyExc1 extends Error { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class C1 {
    public static void main(String [] argv) {
        try {
            p();
        }
        catch( MyExc2 r ) {
        }
    }
    static void p() {
        try {
            System.out.print(1);
            throw( new MyExc3() );
        }
        catch( MyExc3 y ) {
            System.out.print(2);
            throw( new MyExc3() );
        }
    }
}
```

- A. 12Exception in thread "main" MyExc3
  - B. 1222222... (ciclo infinito)
  - C. 12
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

2. Date le dichiarazioni:

```
Integer b;
Boolean g;
Object s;
g = new Boolean(true);
b = new Integer(50);
s = new Integer(50);
```

indicare quali dei seguenti assegnamenti sono corretti a tempo di esecuzione.

- A. b = (Integer) s;
- B. g = (Boolean) s;
- C. b = (Integer) g;

- D. g = (Boolean) b;
  - E. Nessuno dei precedenti
- 

3. Qual è l'output di questo codice?

```
public interface Building {
    int age;
    public int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}
```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

4. Qual'è l'output di questo programma?

```
class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal() {
        super(6);
        System.out.print(5);
    }
}
```

```

class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125  
 B. 135  
 C. 1235  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

5. Qual è l'output di questo codice?

```

public interface Building {
    int getAge();
}

class House implements Building{
    String type = "Residential";
    int age;
    public int getAge() {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione  
 B. Errore a tempo di esecuzione  
 C. 0  
 D. null  
 E. Nessuna delle precedenti
- 

6. Qual'è l'output di questo programma?

```

class X {
    void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    private void f(Y y) {
        System.out.print("Y");
    }
}

class Z extends Y {
    protected void f(Z z) {
        System.out.print("Z");
    }
}

```

```

    }

class Test {
    public static void main(String[] args) {
        X x = new Z();
        Y y = (Y)x;
        Z z = (Z)y;
        z.f(x); z.f(y); z.f(z);
    }
}

```

- A. XXX  
 B. XYZ  
 C. ZZZ  
 D. Errore a tempo di compilazione.  
 E. Nessuna delle precedenti.
- 

7. Dire quale delle seguenti affermazioni è vera:

- A. Un array ha più di una superclasse  
 B. Una classe può essere dichiarata **abstract** o **final**  
 C. Due metodi non possono differire per il tipo di ritorno  
 D. Un array non possiede dei membri  
 E. Ogni valore in memoria non è associato ad un tipo di dato particolare
- 

8. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends Exception { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( MyExc3 b ) {
        }
        catch( MyExc2 j ) {
        }
        catch( Exception v ) {
            System.out.print(2);
        }
        finally {
            throw( new MyExc2() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception g ) {

```

```

        }
    finally {
        System.out.print(3);
    }
}

```

- A. 13  
 B. 1  
 C. 13Exception in thread "main" MyExc2  
 D. Errore a tempo di compilazione  
 E. Nessuna delle precedenti
- 

9. Date le dichiarazioni:

```

Object [] m;
Error [] x;
Boolean [] y;

```

indicare quali dei seguenti assegnamenti sono corretti a tempo di compilazione.

- A. x = (Error []) y;  
 B. x = y;  
 C. y = m;  
 D. x = m;  
 E. m = (Object []) x;
- 

10. Quale output si ottiene invocando il metodo p?

```

class F {
    private String s1 = "";
    private String s2 = s1.concat("a");
    private String s6 = new String("abcd");
    private String s7;
    void p() {
        Boolean b3 = new Boolean(true);
        s7 = s6;
        q(b3, s2, s7, s6);
    }
    void q(Boolean b2, String s3, String s4,
           Object s5) {
        Boolean b1 = new Boolean(true);
        if(b1 == b2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s3 == s2) {
            System.out.print(1);
        } else {
            System.out.print(0);
        }
        if(s4 == s5) {
            System.out.print(1);
        }
    }
}

```

```

    } else {
        System.out.print(0);
    }
}

```

- A. 110  
 B. 101  
 C. 100  
 D. 011  
 E. 010
- 

11. Qual è l'output di questo codice?

```

class MyExc1 extends Exception { }
class MyExc2 extends MyExc1 { }
class MyExc3 extends MyExc2 { }
public class B1 {
    public static void main(String [] argv)
        throws Exception {
        try {
            System.out.print(1);
            m();
        }
        catch( Exception s ) {
            System.out.print(2);
            throw( new MyExc1() );
        }
        catch( MyExc2 z ) {
            System.out.print(3);
        }
        finally {
            throw( new MyExc1() );
        }
    }
    static void m()
        throws Exception {
        try {
        }
        catch( Exception h ) {
        }
        finally {
            System.out.print(4);
        }
    }
}

```

- A. Errore a tempo di compilazione  
 B. 14Exception in thread "main" MyExc1  
 C. 1  
 D. 14222222... (ciclo infinito)  
 E. Nessuna delle precedenti
- 

12. Quanti oggetti *al massimo* possono essere deallocati mediante garbage collection nel punto indicato?

```

public class Coffee {
    String brand ="Nespresso";
    static Integer ml = 20;

    Coffee(){}
    Coffee(int ml){
        this.ml = ml;
    }
    public static void main(String args[]){
        Coffee c1 = new Coffee();
        Coffee c2 = new Coffee();
        Coffee c3 = new Coffee(40);
        c1 = null;
        c3 = c1;
    /* QUI */
        return;
    }
}

```

- A. 1
  - B. 2
  - C. 3
  - D. 4
  - E. 6
- 

13. Dato un file contenente il seguente codice:

```

package pk;

public class O {
    static class I {
    }
    //qui sono i metodi di O...
}

```

Dire quale delle seguenti affermazioni è vera:

- A. Un'istanza della classe I può essere costruita solo all'interno della classe O.
  - B. Un'istanza della classe I può essere costruita solo all'interno del pacchetto pk.
  - C. Un'istanza della classe I può essere costruita solo nelle sottoclassi di O.
  - D. All'esterno del pacchetto pk, un'istanza della classe I può essere costruita con new O.I().
  - E. Nessuna delle precedenti.
- 

14. Qual è l'output di questo codice?

```

public interface Building {
    public int getAge();
}

class House implements Building{
    String type = "Residential";

```

```

    int age;
    public int getAge(int age) {
        return age;
    }
    public static void main(String args[]) {
        House h = new House();
        System.out.println(h.getAge());
    }
}

```

- A. Errore a tempo di compilazione
  - B. Errore a tempo di esecuzione
  - C. 0
  - D. null
  - E. Nessuna delle precedenti
- 

15. Qual è l'output di questo programma?

```

class Animal {
    Animal() {
        System.out.print(2);
    }
    Animal(int i) {
        System.out.print(3);
    }
}
class Mammal extends Animal{
    Mammal(int i) {
        super(6);
        System.out.print(5);
    }
}
class Rabbit extends Mammal {
    public static void main(String args[]) {
        System.out.print(1);
        Mammal m = new Mammal();
    }
}

```

- A. 125
  - B. 135
  - C. 1235
  - D. Errore a tempo di compilazione
  - E. Nessuna delle precedenti
- 

16. Qual è l'output di questo programma?

```

class X {
    private void f(X x) {
        System.out.print("X");
    }
}

class Y extends X {
    void f(Y y) {
        System.out.print("Y");
    }
}

```

```
class Z extends Y {  
    protected void f(Z z) {  
        System.out.print("Z");  
    }  
}  
  
class Test {  
    public static void main(String[] args) {  
        X x = new Z();  
        Y y = new Z();  
        Z z = new Z();  
        z.f(x); z.f(y); z.f(z);  
    }  
}
```

A. XXX  
B. XYZ  
C. ZZZ  
D. Errore a tempo di compilazione.  
E. Nessuna delle precedenti.

---

**Prova n. 1**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 2**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 3**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 4**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 5**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 6**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 8**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 9**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 10**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 11**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 12**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 13**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 15**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**Prova n. 16**

Università di Napoli Federico II – Corso di Laurea in Informatica

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

**LP1**

**Prova d'esame**

*prof. Piero A. Bonatti*

26 giugno 2019

Studente e matricola:

Ora di inizio:

Ora di consegna:

1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
6	A	B	C	D	E
7	A	B	C	D	E
8	A	B	C	D	E
9	A	B	C	D	E
10	A	B	C	D	E
11	A	B	C	D	E
12	A	B	C	D	E
13	A	B	C	D	E
14	A	B	C	D	E
15	A	B	C	D	E
16	A	B	C	D	E

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 1:

1			■				
2				■	■		
3					■	■	
4	■	■					
5			■	■			
6	■	■					
7				■	■		
8					■	■	
9	■	■					
10						■	■
11			■	■			
12					■	■	
13						■	■
14					■	■	
15						■	■
16	■	■					

Risultato prova n. 2:

1			■				
2				■	■		
3			■	■			
4	■	■					
5					■	■	
6					■	■	
7	■	■					
8						■	■
9				■	■		
10	■	■					
11	■	■					
12					■	■	
13				■	■		
14			■	■			
15						■	■
16				■	■		

Risultato prova n. 3:

1		■					
2							■
3				■	■		
4				■	■		
5			■				
6				■	■		
7				■	■		
8			■				
9				■	■		
10				■	■		
11		■	■				
12		■	■				
13		■	■				
14			■				■
15			■				
16					■	■	

Risultato prova n. 4:

## Risultato prova n. 5:

1					
2					
3			■		
4					■
5					■
6	■				
7				■	
8					■
9					■
10	■				
11	■				
12				■	
13	■				
14			■		
15					■
16		■			

Risultato prova n. 6:

1		■					
2							■
3					■	■	
4							■
5			■				
6				■	■		
7				■	■		
8		■					
9						■	
10				■	■		
11		■					
12			■				
13		■					
14			■				
15				■	■		
16						■	

Risultato prova n. 7:

1						
2						
3		■				
4					■	
5			■			
6		■				
7		■				
8						■
9				■		
10				■		
11						■
12				■		
13			■			
14				■		
15			■			
16	■					

Risultato prova n. 8:

1		■					
2				■			
3		■					
4					■		
5						■	
6				■			
7						■	
8				■			
9			■				
10		■					
11			■				
12					■		
13		■					
14					■		
15			■				
16				■			

Risultato prova n. 9:

1		■					
2						■	
3					■	■	
4			■				
5						■	
6				■	■		
7				■	■		
8			■	■			
9			■	■			
10				■	■		
11					■	■	
12					■	■	
13		■	■				
14				■	■		
15		■	■				
16		■	■				

Risultato prova n. 10:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 11:

1			■					
2				■				
3					■			
4						■		
5							■	
6								■
7								■
8		■						
9			■					
10								■
11						■		
12					■			
13				■				
14		■						
15			■					
16								■

Risultato prova n. 12:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 13:

1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Risultato prova n. 14:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 15:

1		■					
2			■				
3		■					
4						■	
5			■				
6			■				
7				■			
8					■		
9				■			
10		■					
11		■					
12			■				
13					■		
14					■		
15					■		
16			■				

Risultato prova n. 16:

1		■					
2				■	■		
3						■	■
4			■	■			
5					■	■	
6							■
7						■	■
8		■	■				
9					■	■	
10		■	■				
11					■	■	
12						■	■
13			■	■			
14			■	■			
15		■	■				
16							■

Risultato prova n. 17:

1		■					
2			■				
3				■			
4				■			
5					■		
6		■					
7		■					
8			■				
9				■			
10				■			
11					■		
12		■					
13				■			
14					■		
15						■	
16						■	

Risultato prova n. 18:

1																	
2																	
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	
16																	

Risultato prova n. 19:

1			■				
2					■	■	
3						■	■
4				■			
5					■	■	
6	■	■					
7					■	■	
8			■	■			
9			■	■			
10	■	■					
11					■		
12	■	■					
13	■	■					
14				■	■		
15				■	■		
16						■	

Risultato prova n. 20:

1		■					
2				■	■		
3			■	■			
4					■	■	
5		■	■				
6			■	■			
7			■	■			
8					■	■	
9						■	■
10					■	■	
11						■	■
12	■	■					
13							■
14	■	■					
15					■	■	
16						■	■

Risultato prova n. 21:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 22:

1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Risultato prova n. 23:

1			■				
2	■						
3					■	■	
4			■	■			
5			■	■			
6	■	■					
7	■	■					
8	■						
9					■	■	
10				■	■		
11				■	■		
12				■	■		
13				■	■		
14						■	■
15						■	■
16					■		

Risultato prova n. 24:

1			■				
2					■	■	
3					■	■	
4				■			
5						■	
6	■	■					
7			■	■			
8	■	■					
9				■	■		
10			■	■			
11					■	■	
12				■	■		
13				■	■		
14						■	
15	■	■					
16	■	■					

Risultato prova n. 25:

1		■					
2			■				
3		■					
4				■			
5			■				
6					■		
7		■					
8				■			
9		■					
10					■		
11						■	
12				■			
13						■	
14					■		
15			■				
16				■			

Risultato prova n. 26:

1		■					
2				■			
3							■
4					■		
5						■	
6		■					
7						■	
8			■				
9				■			
10		■					
11			■				
12				■			
13						■	
14				■			
15					■		
16		■					

Risultato prova n. 27:

1		■					
2			■				
3		■					
4				■			
5					■		
6					■		
7		■					
8		■					
9				■			
10			■				
11				■			
12			■				
13					■		
14					■		
15					■		
16			■				

Risultato prova n. 28:

1					
2	■				
3		■			
4					■
5			■		
6		■			
7		■			
8				■	
9	■				
10				■	
11			■		
12			■		
13	■				
14					■
15				■	
16	■				

Risultato prova n. 29:

1		■					
2					■	■	
3			■	■			
4					■	■	
5			■	■			
6					■	■	
7						■	■
8					■	■	
9						■	■
10		■	■				
11			■	■			
12	■	■					
13						■	■
14				■	■		
15	■	■					
16					■	■	

Risultato prova n. 30:

1		■					
2			■				
3						■	
4				■			
5						■	
6	■						
7				■			
8				■			
9				■			
10	■						
11					■		
12			■				
13	■						
14					■		
15			■				
16					■		

Risultato prova n. 31:

## Risultato prova n. 32:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 33:

A 16x16 grid containing several red squares. The positions of the red squares are as follows:

- (1, 15)
- (2, 2)
- (2, 15)
- (3, 2)
- (3, 15)
- (4, 15)
- (5, 2)
- (5, 15)
- (6, 2)
- (6, 15)
- (7, 15)
- (8, 2)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 2)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 2)
- (15, 15)
- (16, 2)

## Risultato prova n. 34:

1			■				
2				■	■		
3				■	■		
4						■	■
5	■	■					
6						■	■
7	■	■					
8	■						
9					■	■	
10			■	■			
11				■	■		
12				■	■		
13					■	■	
14			■	■			
15					■	■	
16	■	■					

Risultato prova n. 35:

1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Risultato prova n. 36:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 37:

1						1
2			1			2
3					1	3
4						1
5	1	1				
6	1	1				
7					1	7
8	1	1				
9	1	1				
10				1	1	
11			1			
12					1	
13			1			
14				1	1	
15				1	1	
16				1		

Risultato prova n. 38:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 39:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 40:

A 16x16 grid with red squares at specific intersections. The red squares are located at the following coordinates (row, column): (1, 16), (2, 3), (3, 16), (4, 2), (5, 2), (6, 6), (7, 2), (8, 8), (9, 6), (10, 5), (11, 6), (12, 2), (13, 5), (14, 8), (15, 8), and (16, 4).

## Risultato prova n. 41:

1			■				
2				■	■		
3				■	■		
4						■	■
5				■	■		
6			■	■			
7	■	■					
8	■						
9					■	■	
10				■	■		
11					■	■	
12						■	■
13			■	■			
14	■	■					
15						■	■
16	■	■					

Risultato prova n. 42:

1							1
2							2
3							3
4							4
5							5
6							6
7							7
8							8
9							9
10							10
11							11
12							12
13							13
14							14
15							15
16							16

Risultato prova n. 43:

1			■		
2				■	
3					■
4	■				
5				■	
6		■			
7					■
8				■	
9			■		
10	■				
11	■				
12			■		
13			■		
14	■				
15			■		
16		■			

Risultato prova n. 44:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 45:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 46:

1			■		
2		■			
3			■		
4				■	
5	■				
6					■
7				■	
8		■			
9			■		
10	■				
11	■				
12	■				
13			■		
14				■	
15					■
16		■			

Risultato prova n. 47:

1			■				
2				■			
3					■		
4						■	
5						■	
6				■			
7	■						
8			■				
9	■						
10	■						
11				■			
12					■		
13			■				
14						■	
15				■			
16	■						

Risultato prova n. 48:

A 16x16 grid with red squares at the following coordinates:

- (1, 1)
- (2, 1)
- (3, 1)
- (4, 1)
- (5, 1)
- (6, 1)
- (7, 1)
- (8, 1)
- (9, 1)
- (10, 1)
- (11, 1)
- (12, 1)
- (13, 1)
- (14, 1)
- (15, 1)
- (16, 1)
- (1, 2)
- (2, 2)
- (3, 2)
- (4, 2)
- (5, 2)
- (6, 2)
- (7, 2)
- (8, 2)
- (9, 2)
- (10, 2)
- (11, 2)
- (12, 2)
- (13, 2)
- (14, 2)
- (15, 2)
- (16, 2)
- (1, 3)
- (2, 3)
- (3, 3)
- (4, 3)
- (5, 3)
- (6, 3)
- (7, 3)
- (8, 3)
- (9, 3)
- (10, 3)
- (11, 3)
- (12, 3)
- (13, 3)
- (14, 3)
- (15, 3)
- (16, 3)
- (1, 4)
- (2, 4)
- (3, 4)
- (4, 4)
- (5, 4)
- (6, 4)
- (7, 4)
- (8, 4)
- (9, 4)
- (10, 4)
- (11, 4)
- (12, 4)
- (13, 4)
- (14, 4)
- (15, 4)
- (16, 4)
- (1, 5)
- (2, 5)
- (3, 5)
- (4, 5)
- (5, 5)
- (6, 5)
- (7, 5)
- (8, 5)
- (9, 5)
- (10, 5)
- (11, 5)
- (12, 5)
- (13, 5)
- (14, 5)
- (15, 5)
- (16, 5)
- (1, 6)
- (2, 6)
- (3, 6)
- (4, 6)
- (5, 6)
- (6, 6)
- (7, 6)
- (8, 6)
- (9, 6)
- (10, 6)
- (11, 6)
- (12, 6)
- (13, 6)
- (14, 6)
- (15, 6)
- (16, 6)
- (1, 7)
- (2, 7)
- (3, 7)
- (4, 7)
- (5, 7)
- (6, 7)
- (7, 7)
- (8, 7)
- (9, 7)
- (10, 7)
- (11, 7)
- (12, 7)
- (13, 7)
- (14, 7)
- (15, 7)
- (16, 7)
- (1, 8)
- (2, 8)
- (3, 8)
- (4, 8)
- (5, 8)
- (6, 8)
- (7, 8)
- (8, 8)
- (9, 8)
- (10, 8)
- (11, 8)
- (12, 8)
- (13, 8)
- (14, 8)
- (15, 8)
- (16, 8)
- (1, 9)
- (2, 9)
- (3, 9)
- (4, 9)
- (5, 9)
- (6, 9)
- (7, 9)
- (8, 9)
- (9, 9)
- (10, 9)
- (11, 9)
- (12, 9)
- (13, 9)
- (14, 9)
- (15, 9)
- (16, 9)
- (1, 10)
- (2, 10)
- (3, 10)
- (4, 10)
- (5, 10)
- (6, 10)
- (7, 10)
- (8, 10)
- (9, 10)
- (10, 10)
- (11, 10)
- (12, 10)
- (13, 10)
- (14, 10)
- (15, 10)
- (16, 10)
- (1, 11)
- (2, 11)
- (3, 11)
- (4, 11)
- (5, 11)
- (6, 11)
- (7, 11)
- (8, 11)
- (9, 11)
- (10, 11)
- (11, 11)
- (12, 11)
- (13, 11)
- (14, 11)
- (15, 11)
- (16, 11)
- (1, 12)
- (2, 12)
- (3, 12)
- (4, 12)
- (5, 12)
- (6, 12)
- (7, 12)
- (8, 12)
- (9, 12)
- (10, 12)
- (11, 12)
- (12, 12)
- (13, 12)
- (14, 12)
- (15, 12)
- (16, 12)
- (1, 13)
- (2, 13)
- (3, 13)
- (4, 13)
- (5, 13)
- (6, 13)
- (7, 13)
- (8, 13)
- (9, 13)
- (10, 13)
- (11, 13)
- (12, 13)
- (13, 13)
- (14, 13)
- (15, 13)
- (16, 13)
- (1, 14)
- (2, 14)
- (3, 14)
- (4, 14)
- (5, 14)
- (6, 14)
- (7, 14)
- (8, 14)
- (9, 14)
- (10, 14)
- (11, 14)
- (12, 14)
- (13, 14)
- (14, 14)
- (15, 14)
- (16, 14)
- (1, 15)
- (2, 15)
- (3, 15)
- (4, 15)
- (5, 15)
- (6, 15)
- (7, 15)
- (8, 15)
- (9, 15)
- (10, 15)
- (11, 15)
- (12, 15)
- (13, 15)
- (14, 15)
- (15, 15)
- (16, 15)
- (1, 16)
- (2, 16)
- (3, 16)
- (4, 16)
- (5, 16)
- (6, 16)
- (7, 16)
- (8, 16)
- (9, 16)
- (10, 16)
- (11, 16)
- (12, 16)
- (13, 16)
- (14, 16)
- (15, 16)
- (16, 16)

## Risultato prova n. 49:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 50:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 51:

1		■					
2				■	■		
3				■	■		
4						■	■
5			■	■			
6					■	■	
7							■
8		■	■				
9		■	■				
10					■	■	
11				■	■		
12			■	■			
13			■	■			
14				■	■		
15							■
16	■	■					

Risultato prova n. 52:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 53:

1					
2	■				
3		■			
4			■		
5					■
6				■	■
7				■	■
8			■		
9	■				
10		■			
11				■	
12	■				
13			■		
14	■				
15		■			
16					■

Risultato prova n. 54:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 55:

1		■					
2				■			
3		■					
4						■	
5					■		
6			■				
7				■			
8						■	
9					■		
10				■			
11			■				
12	■						
13				■			
14	■						
15			■				
16					■		

Risultato prova n. 56:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 57:

1		■					
2							■
3					■	■	
4	■						
5				■	■		
6	■						
7			■				
8							■
9				■	■		
10						■	
11				■	■		
12			■	■			
13				■	■		
14	■						
15						■	
16		■					

Risultato prova n. 58:

1			■				
2						■	■
3					■	■	
4						■	■
5	■	■					
6	■	■					
7			■	■			
8				■	■		
9	■	■					
10				■	■		
11	■	■					
12					■	■	
13			■	■			
14				■	■		
15				■	■		
16					■	■	

Risultato prova n. 59:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 60:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 61:

## Risultato prova n. 62:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 63:

1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Risultato prova n. 64:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 65:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 66:

1			■				
2					■	■	
3						■	■
4	■	■					
5	■	■					
6					■	■	
7						■	■
8				■	■		
9	■	■					
10				■	■		
11					■	■	
12	■	■					
13				■	■		
14			■	■			
15				■	■		
16		■	■				

Risultato prova n. 67:

1		■					
2						■	
3				■	■		
4					■		
5						■	■
6						■	■
7		■	■				
8		■					
9			■	■			
10			■	■			
11						■	
12		■	■				
13							■
14			■	■			
15					■		
16					■		

Risultato prova n. 68:

## Risultato prova n. 69:

1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Risultato prova n. 70:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 71:

1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Risultato prova n. 72:

### **Versione 1 dell'esercizio 1**

Dire quale delle seguenti affermazioni è falsa:

- A) In Java tutti i tipi numerici sono con segno
- B) Una classe può essere dichiarata **final**
- C) Un array possiede dei membri
- D) Una classe può essere dichiarata **abstract** o **final**
- E) Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri

## **Versione 2 dell'esercizio 1**

Dire quale delle seguenti affermazioni è falsa:

- A) La dichiarazione di un oggetto e la sua creazione possono essere svolte in tempi diversi
- B) L'operatore `new` restituisce un riferimento all'oggetto appena creato
- C) Una classe può essere dichiarata `abstract` o `final`
- D) Una classe può essere dichiarata `private`
- E) Una classe non interna può essere dichiarata `private`

### **Versione 3 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) Un attributo **static** non può essere acceduto mediante il nome della sua classe di appartenenza
- B) Una classe non può essere dichiarata **private**
- C) Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- D) I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- E) La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente

#### **Versione 4 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) Un oggetto esiste sempre dopo la sua dichiarazione
- B) I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi
- C) Nell'overloading due metodi non possono avere lo stesso nome ma diverso tipo di parametri
- D) Un attributo **static** non può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- E) Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

### **Versione 5 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) La lunghezza di un array può essere variata dopo la sua costruzione
- B) Un array ha un'unica superclasse
- C) Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- D) Un array non possiede dei membri
- E) La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente

### **Versione 6 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) I modificatori che precedono una variabile di tipo array si applicano alla variabile array ed anche ai suoi elementi
- B) Ogni valore in memoria non è associato ad un tipo di dato particolare
- C) Ai metodi **static** non si applica il *dynamic method dispatch*
- D) L'operatore **new** non restituisce un riferimento all'oggetto appena creato
- E) Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri

### **Versione 7 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) La lunghezza di un array può essere variata dopo la sua costruzione
- B) Una classe non interna non può essere dichiarata `private`
- C) Un array non possiede dei membri
- D) Una classe non può essere dichiarata `abstract` o `final`
- E) La dimensione di un array deve essere indicata al momento della dichiarazione dell'array

### **Versione 8 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) Non tutti i tipi di eccezioni estendono la classe `Object`
- B) La dichiarazione di un oggetto e la sua creazione possono essere svolte solo in tempi diversi
- C) Un attributo non può essere contemporaneamente `static` e `private`
- D) In Java non tutti i tipi numerici sono con segno
- E) Un array vuoto non può avere riferimento `null`

### **Versione 9 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) Due metodi non possono avere diverso nome ma lo stesso tipo di parametri
- B) Ai metodi **static** si applica il *dynamic method dispatch*
- C) Un attributo non può essere contemporaneamente **static** e **private**
- D) Un attributo **static** può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- E) Un attributo non può essere contemporaneamente **static** e **final**

### **Versione 10 dell'esercizio 1**

Dire quale delle seguenti affermazioni è falsa:

- A) Un attributo **static** può essere acceduto mediante il nome della sua classe di appartenenza
- B) Un attributo **static** può essere acceduto mediante un riferimento a un oggetto della sua classe di appartenenza
- C) Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri
- D) Il minimo numero di elementi che un array può contenere è 0
- E) Un attributo può essere contemporaneamente **static** e **final**

### **Versione 11 dell'esercizio 1**

Dire quale delle seguenti affermazioni è falsa:

- A) Non è possibile dichiarare un attributo senza inizializzarlo
- B) Ogni valore in memoria è associato ad un tipo di dato particolare
- C) Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default
- D) La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente
- E) In Java tutti i tipi numerici sono con segno

### **Versione 12 dell'esercizio 1**

Dire quale delle seguenti affermazioni è falsa:

- A) Non è possibile dichiarare un attributo senza inizializzarlo
- B) La dimensione di un array può non essere indicata al momento della dichiarazione dell'array
- C) Quando due metodi hanno lo stesso nome non sempre si ha overloading
- D) La dichiarazione di un oggetto e la sua creazione possono essere svolte contemporaneamente
- E) Prima dell'esecuzione del costruttore alle variabili di istanza sono assegnati i valori di default

### **Versione 13 dell'esercizio 1**

Dire quale delle seguenti affermazioni è falsa:

- A) Un array ha un'unica superclasse
- B) Quando due metodi hanno lo stesso nome non sempre si ha overloading
- C) Un array vuoto può avere riferimento `null`
- D) Ogni valore in memoria è associato ad un tipo di dato particolare
- E) I modificatori applicati a una variabile di tipo array si applicano alla variabile array e non ai suoi elementi

### **Versione 14 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) Il minimo numero di elementi che un array può contenere è 1
- B) Una classe non può essere dichiarata `protected`
- C) Un array non possiede dei membri
- D) È possibile dichiarare un array senza indicarne la dimensione
- E) Non è possibile dichiarare un attributo senza inizializzarlo

### **Versione 15 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) Non è possibile dichiarare un attributo senza inizializzarlo
- B) Non tutti i tipi di eccezioni estendono la classe `Throwable`
- C) Un array vuoto non può avere riferimento `null`
- D) La lunghezza di un array può essere variata dopo la sua costruzione
- E) Due metodi non possono avere diverso nome ma lo stesso tipo di parametri

### **Versione 16 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) Un attributo può essere contemporaneamente **static** e **private**
- B) Un attributo non può essere contemporaneamente **static** e **final**
- C) Il minimo numero di elementi che un array può contenere è 1
- D) La lunghezza di un array può essere variata dopo la sua creazione
- E) Nell'overloading due metodi non possono avere lo stesso nome e lo stesso numero di parametri

### **Versione 17 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) Un array non possiede dei membri
- B) Quando due metodi hanno lo stesso nome si ha overloading
- C) Un array con riferimento `null` non ha lunghezza
- D) Un attributo non può essere contemporaneamente `static` e `private`
- E) I modificatori applicati a una variabile di tipo array non si applicano alla variabile array ma ai suoi elementi

### **Versione 18 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) Non è possibile dichiarare un array senza indicarne la dimensione
- B) Non tutti i tipi di eccezioni estendono la classe `Object`
- C) Il minimo numero di elementi che un array può contenere è 0
- D) La dichiarazione di un oggetto e la sua creazione possono essere svolte solo contemporaneamente
- E) Una classe non può essere dichiarata `private`

### **Versione 19 dell'esercizio 1**

Dire quale delle seguenti affermazioni è vera:

- A) Una classe non può essere dichiarata `protected`
- B) Una classe può essere dichiarata `private`
- C) Non tutti i tipi di eccezioni estendono la classe `Object`
- D) Un array vuoto può avere riferimento `null`
- E) Ogni valore in memoria non è associato ad un tipo di dato particolare