

**Oracle 1z0-808**



**Java SE 8 Programmer I**

**Version: 8.0**

**QUESTION NO: 1**

Given:

```
class Product {  
    double price;  
}  
  
public class Test {  
    public void updatePrice(Product product, double price) {  
        price = price * 2;  
        product.price = product.price + price;  
    }  
    public static void main(String[] args) {  
        Product prt = new Product();  
        prt.price = 200;  
        double newPrice = 100;  
  
        Test t = new Test();  
        t.updatePrice(prt, newPrice);  
        System.out.println(prt.price + " : " + newPrice);  
    }  
}
```

What is the result?

**A.**

200.0 : 100.0

**B.**

400.0 : 200.0

**C.**

400.0 : 100.0

**D.**

Compilation fails.

**Answer: C**

**Explanation:**

**QUESTION NO: 2**

Given the code fragment:

```
if (aVar++ < 10) {  
    System.out.println(aVar + " Hello World!");  
} else {  
    System.out.println(aVar + " Hello Universe!");  
}
```

What is the result if the integer aVar is 9?

**A.**

10 Hello World!

**B.**

Hello Universe!

**C.**

Hello World!

**D.**

Compilation fails.

**Answer:** A

**Explanation:**

### QUESTION NO: 3

Given the code fragment:

```
public static void main(String[] args) {  
    String date = LocalDate  
        .parse("2014-05-04")  
        .format(DateTimeFormatter.ISO_DATE_TIME);  
    System.out.println(date);  
}
```

What is the result?

**A.**

May 04, 2014T00:00:00.000

**B.**

2014-05-04T00:00: 00. 000

**C.**

5/4/14T00:00:00.000

**D.**

An exception is thrown at runtime.

**Answer: D**

**Explanation:**

#### QUESTION NO: 4

Given the code fragment:

```
public static void main(String[] args) {  
    Short s1 = 200;  
    Integer s2 = 400;  
    Long s3 = (long) s1 + s2;           //line n1  
    String s4 = (String) (s3 * s2);    //line n2  
    System.out.println("Sum is " + s4);  
}
```

What is the result?

**A.**

Sum is 600

**B.**

Compilation fails at line n1.

**C.**

Compilation fails at line n2.

**D.**

A ClassCastException is thrown at line n1.

**E.**

A ClassCastException is thrown at line n2.

**Answer: C**

**Explanation:**

#### QUESTION NO: 5

What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?

- A.**  
Encapsulation
- B.**  
Inheritance
- C.**  
Abstraction
- D.**  
Instantiation
- E.**  
Polymorphism

**Answer: A**

**Explanation:**

Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

Reference:[http://www.tutorialspoint.com/java/java\\_access\\_modifiers.htm](http://www.tutorialspoint.com/java/java_access_modifiers.htm)

## QUESTION NO: 6

Given the code fragment:

```
abstract class Planet {  
    protected void revolve() { //line n1  
    }  
  
    abstract void rotate(); //line n2  
}  
  
class Earth extends Planet {  
    void revolve() { //line n3  
    }  
  
    protected void rotate() { //line n4  
    }  
}
```

Which two modifications, made independently, enable the code to compile?

- A.**  
Make the method at line n1 public.

**B.**

Make the method at line n2 public.

**C.**

Make the method at line n3 public.

**D.**

Make the method at line n3 protected.

**E.**

Make the method at line n4 public.

**Answer: C,D**

**Explanation:**

## QUESTION NO: 7

Given:

```
class Vehicle {  
    String type = "4W";  
    int maxSpeed = 100;  
  
    Vehicle(String type, int maxSpeed) {  
        this.type = type;  
        this.maxSpeed = maxSpeed;  
    }  
}  
  
class Car extends Vehicle {  
    String trans;  
  
    Car(String trans) {           //line n1  
        this.trans = trans;  
    }  
  
    Car(String type, int maxSpeed, String trans) {  
        super(type, maxSpeed);  
        this(trans);           //line n2  
    }  
}
```

And given the code fragment:

```
7. Car c1 = new Car("Auto");
8. Car c2 = new Car("4W", 150, "Manual");
9. System.out.println(c1.type + " " + c1.maxSpeed + " " + c1.trans);
10. System.out.println(c2.type + " " + c2.maxSpeed + " " + c2.trans);
```

What is the result?

**A.**

4W 100 Auto

4W 150 Manual

**B.**

Null 0 Auto

4W 150 Manual

**C.**

Compilation fails only at line n1

**D.**

Compilation fails only at line n2

**E.**

Compilation fails at both line n1 and line n2

**Answer: C**

**Explanation:**

## QUESTION NO: 8

fragment:

```
1. class X {
2.     public void printFileContent() {
3.         /* code goes here */
4.         throw new IOException();
5.     }
6. }
7. public class Test {
8.     public static void main(String[] args) {
9.         X xobj = new X();
10.        xobj.printFileContent();
11.    }
12. }
```

Which two modifications should you make so that the code compiles successfully?

- A) Replace line 8 with public static void main(String[] args) throws Exception {
- B) Replace line 10 with:

```
try {
    xobj.printFileContent();
}
catch(Exception e) {
}
catch(IOException e) { }
```
- C) Replace line 2 with public void printFileContent() throws IOException {
- D) Replace line 4 with throw IOException("Exception raised");
- E) At line 11, insert throw new IOException();

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**E.**

Option E

**Answer:** A,C

**Explanation:**

### QUESTION NO: 9

Given the following two classes:

```
public class Customer {
    ElectricAccount acct = new ElectricAccount();

    public void useElectricity(double kWh) {
        acct.addKWh(kWh);
    }
}

public class ElectricAccount {
    private double kWh;
    private double rate = 0.07;
    private double bill;

    //line n1
}
```

How should you write methods in the ElectricAccount class at line n1 so that the member variable bill is always equal to the value of the member variable kwh multiplied by the member variable rate?

Any amount of electricity used by a customer (represented by an instance of the customer class) must contribute to the customer's bill (represented by the member variable bill) through the method use Electricity method. An instance of the customer class should never be able to tamper with or decrease the value of the member variable bill.

- A) 

```
public void addKWh(double kWh) {  
    this.kWh += kWh;  
    this.bill = this.kWh*this.rate;  
}
```
- B) 

```
public void addKWh(double kWh) {  
    if (kWh > 0){  
        this.kWh += kWh;  
        this.bill = this.kWh * this.rate;  
    }  
}
```
- C) 

```
private void addKWh(double kWh) {  
    if (kWh > 0) {  
        this.kWh += kWh;  
        this.bill = this.kWh*this.rate;  
    }  
}
```
- D) 

```
public void addKWh(double kWh) {  
    if(kWh > 0) {  
        this.kWh += kWh;  
        setBill(this.kWh);  
    }  
}  
public void setBill(double kWh) {  
    bill = kWh*rate;  
}
```

A.

Option A

B.

Option B

C.

Option C

D.

Option D

**Answer: A,C**

**Explanation:**

**QUESTION NO: 10**

Given the code fragment:

```
public static void main(String[] args) {  
    StringBuilder sb = new StringBuilder(5);  
    String s = "";  
  
    if (sb.equals(s)) {  
        System.out.println("Match 1");  
    } else if (sb.toString().equals(s.toString())) {  
        System.out.println("Match 2");  
    } else {  
        System.out.println("No Match");  
    }  
}
```

What is the result?

**A.**

Match 1

**B.**

Match 2

**C.**

No Match

**D.**

A NullPointerException is thrown at runtime.

**Answer: B**

**Explanation:**

**QUESTION NO: 11**

Given:

```
interface Readable {  
    public void readBook();  
    public void setBookMark();  
}  
  
abstract class Book implements Readable { // line n1  
    public void readBook() {}  
    // line n2  
}  
  
class EBook extends Book { // line n3  
    public void readBook() {}  
    // line n4  
}
```

Which option enables the code to compile?

- A) Replace the code fragment at line n1 with:  
 class Book implements Readable {
- B) At line n2 insert:  
 public abstract void setBookMark();
- C) Replace the code fragment at line n3 with:  
 abstract class EBook extends Book {
- D) At line n4 insert:  
 public void setBookMark() {}

A.

Option A

B.

Option B

C.

Option C

D.

Option D

**Answer: C**

**Explanation:**

**QUESTION NO: 12**

Given:

```
public static void main(String[] args) {  
    String ta = "A ";  
    ta = ta.concat("B ");  
    String tb = "C ";  
    ta = ta.concat(tb);  
    ta.replace('C', 'D');  
    ta = ta.concat(tb);  
    System.out.println(ta);  
}
```

What is the result?

**A.**

A B C D

**B.**

A C D

**C.**

A B C C

**D.**

A B D

**E.**

A B D C

**Answer: E**

**Explanation:**

### QUESTION NO: 13

Given:

```
class CD {  
    int r;  
    CD(int r) {  
        this.r=r;  
    }  
}  
  
class DVD extends CD {  
    int c;  
    DVD(int r, int c) {  
        // line n1  
    }  
}
```

And given the code fragment:

```
DVD dvd = new DVD(10, 20);
```

Which code fragment should you use at line n1 to instantiate the dvd object successfully?

- A) super.r = r;  
 this.c = c;
- B) super(r);  
 this(c);
- C) super(r);  
 this.c = c;
- D) this.c = r;  
 super(c);

A.

Option A

B.

Option B

C.

Option C

D.

Option D

**Answer: C**

**Explanation:**

**QUESTION NO: 14**

Given the code fragment:

```
int a[] = {1, 2, 3, 4, 5};  
for(XXX) {  
    System.out.print(a[e]);  
}
```

Which option can replace xxx to enable the code to print 135?

**A.**

int e = 0; e <= 4; e++

**B.**

int e = 0; e < 5; e += 2

**C.**

int e = 1; e <= 5; e += 1

**D.**

int e = 1; e < 5; e+ =2

**Answer: B**

**Explanation:**

**QUESTION NO: 15**

Which statement best describes encapsulation?

**A.**

Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects.

**B.**

Encapsulation ensures that classes can be designed so that their methods are inheritable.

**C.**

Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.

D.

Encapsulation ensures that classes can be designed so that if a method has an argument MyType x, any subclass of MyType can be passed to that method.

**Answer: A**

**Explanation:**

#### QUESTION NO: 16

Given the code fragment from three files:

SalesMan.java:

```
package sales;
public class SalesMan { }
```

Product.java:

```
package sales.products;
public class Product { }
```

Market.java:

```
1. package market;
2. // insert code here
3. public class USMarket {
4.     SalesMan sm;
5.     Product p;
6. }
```

Which code fragment, when inserted at line 2, enables the code to compile?

- A) import sales.\*;
- B) import java.sales.products.\*;
- C) import sales;  
    import sales.products;
- D) import sales.\*;  
    import products.\*;
- E) import sales.\*;  
    import sales.products.\*;

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**E.**

Option E

**Answer: E**

**Explanation:**

#### **QUESTION NO: 17**

Given the following class:

```
public class CheckingAccount {  
    public int amount;  
    public CheckingAccount(int amount) {  
        this.amount = amount;  
    }  
    public int getAmount () {  
        return amount;  
    }  
    public void changeAmount (int x) {  
        amount += x;  
    }  
}
```

And given the following main method, located in another class:

```
public static void main(String[] args) {  
    CheckingAccount acct = new CheckingAccount ((int) (Math.random() *1000));  
    //line n1  
    System.out.println(acct.getAmount());  
}
```

Which three lines, when inserted independently at line n1, cause the program to print a 0 balance?

**A.**

this.amount = 0;

**B.**

amount = 0;

**C.**

acct(0);

**D.**

acct.amount = 0;

**E.**

acct.getAmount () = 0;

**F.**

acct.changeAmount(0);

**G.**

acct.changeAmount(-acct.amount);

**H.**

acct.changeAmount(-acct.getAmount());

**Answer: A,C,D**

**Explanation:****QUESTION NO: 18**

Given the code fragment:

```
String shirts[][] = new String[2][2];
shirts[0][0] = "red";
shirts[0][1] = "blue";
shirts[1][0] = "small";
shirts[1][1] = "medium";
```

Which code fragment prints red: blue: small: medium?

- A) 

```
for (int index = 1; index < 2; index++) {
    for (int idx = 1; idx < 2; idx++) {
        System.out.print(shirts[index][idx] + ":");
    }
}
```
- B) 

```
for (int index = 0; index < 2; ++index) {
    for (int idx = 0; idx < index; ++idx) {
        System.out.print(shirts[index][idx] + ":");
    }
}
```
- C) 

```
for (String c : colors) {
    for (String s : sizes) {
        System.out.println(s + ":");
    }
}
```
- D) 

```
for (int index = 0; index < 2;) {
    for (int idx = 0; idx < 2;) {
        System.out.print(shirts[index][idx] + ":");
        idx++;
    }
    index++;
}
```

**A.**

Option A

**B.**

Option B

**C.**

Option C

D.

Option D

**Answer: D**

**Explanation:**

### QUESTION NO: 19

Given the code fragment:

```
public class Test{  
    void readCard(int cardNo) throws Exception {  
        System.out.println("Reading Card");  
    }  
  
    void checkCard(int cardNo) throws RuntimeException { // line n1  
        System.out.println("Checking Card");  
    }  
  
    public static void main(String[] args) {  
        Test ex = new Test();  
        int cardNo = 12344;  
        ex.checkCard(cardNo);  
        ex.readCard(cardNo);  
    }  
}
```

What is the result?

A.

Reading Card

Checking Card

B.

Compilation fails only at line n1.

C.

Compilation fails only at line n2.

D.

Compilation fails only at line n3.

E.

Compilation fails at both line n2 and line n3.

**Answer: D**

**Explanation:**

## QUESTION NO: 20

Given the code fragment:

```
3. public static void main(String[] args) {  
4.     int x = 5;  
5.     while (isAvailable(x)) {  
6.         System.out.print(x);  
7.     }  
8. }  
10.  
11. public static boolean isAvailable(int x) {  
12.     return x-- > 0 ? true : false;  
13. }
```

Which modification enables the code to print 54321?

**A.**

Replace line 6 with System.out.print(--x);

**B.**

At line7, insert x--;

**C.**

Replace line 6 with --x; and, at line 7, insert system.out.print(x);

**D.**

Replace line 12 With return (x > 0) ? false: true;

**Answer: A**

**Explanation:**

## QUESTION NO: 21

Given the code fragment:

```
4. public static void main(String[] args) {  
5.     boolean opt = true;  
6.     switch (opt) {  
7.         case true:  
8.             System.out.print("True");  
9.             break;  
10.        default:  
11.            System.out.print("****");  
12.    }  
13.    System.out.println("Done");  
14. }
```

Which modification enables the code fragment to print TrueDone?

**A.**

Replace line 5 With String opt= "true";

Replace line 7 with case "true":

**B.**

Replace line 5 with boolean opt = l;

Replace line 7 with case 1=

**C.**

At line 9, remove the break statement.

**D.**

Remove the default section.

**Answer: A**

**Explanation:**

## QUESTION NO: 22

Given the following main method:

```
public static void main(String[] args) {  
    int num = 5;  
    do {  
        System.out.print(num-- + " ");  
    } while(num == 0);  
}
```

What is the result?

**A.**  
5 4 3 2 1 0

**B.**  
5 4 3 2 1

**C.**  
4 2 1

**D.**  
5

**E.**  
Nothing is printed

**Answer:** D

**Explanation:**

### QUESTION NO: 23

Given the code fragment:

```
int x = 100;
int a = x++;
int b = ++x;
int c = x++;
int d = (a < b) ? (a < c) ? a: (b < c )? b: c;
System.out.println(d);
```

What is the result?

**A.**  
100

**B.**  
101

**C.**  
102

**D.**  
103

**E.**

Compilation fails

**Answer: E****Explanation:****QUESTION NO: 24**

Given:

```
public class Test {  
  
    public static void main(String[] args) {  
  
        String[][] chs = new String[2][];  
        chs[0] = new String[2];  
        chs[1] = new String[5];  
        int i = 97;  
  
        for (int a = 0; a < chs.length; a++) {  
            for (int b = 0; b < chs.length; b++) {  
                chs[a][b] = "" + i;  
                i++;  
            }  
        }  
  
        for (String[] ca : chs) {  
            for (String c : ca) {  
                System.out.print(c + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

What is the result?

**A.**

97 98

99 100 null null null

**B.**

97 98

99 100 101 102 103

C.

Compilation rails.

D.

A NullPointerException is thrown at runtime.

E.

An ArrayIndexOutOfBoundsException is thrown at runtime.

**Answer:** A

**Explanation:**

### QUESTION NO: 25

Given the code fragment:

```
public class Employee {  
    String name;  
    boolean contract;  
    double salary;  
    Employee() {  
        // line n1  
    }  
    public String toString(){  
        return name + ":" + contract + ":" + salary;  
    }  
    public static void main(String[] args) {  
        Employee e = new Employee();  
        // line n2  
        System.out.print(e);  
    }  
}
```

Which two modifications, when made independently, enable the code to print joe:true: 100.0?

- A) Replace line n2 with:  

```
e.name = "Joe";  
e.contract = true;  
e.salary = 100;
```
- B) Replace line n2 with:  

```
this.name = "Joe";  
this.contract = true;  
this.salary = 100;
```
- C) Replace line n1 with:  

```
this.name = new String("Joe");  
this.contract = new Boolean(true);  
this.salary = new Double(100);
```
- D) Replace line n1 with:  

```
name = "Joe";  
contract = TRUE;  
salary = 100.0f;
```
- E) Replace line n1 with:  

```
this("Joe", true, 100);
```

A.

Option A

B.

Option B

C.

Option C

D.

Option D

E.

Option E

**Answer: A,C**

**Explanation:**

**QUESTION NO: 26**

Given the code fragment:

```
public static void main(String[] args) {  
    List<String> names = new ArrayList<>();  
    names.add("Robb");  
    names.add("Bran");  
    names.add("Rick");  
    names.add("Bran");  
  
    if (names.remove("Bran")) {  
        names.remove("Jon");  
    }  
    System.out.println(names);  
}
```

What is the result?

**A.**

[Robb, Rick, Bran]

**B.**

[Robb, Rick]

**C.**

[Robb, Bran, Rick, Bran]

**D.**

An exception is thrown at runtime.

**Answer: A**

**Explanation:**

## QUESTION NO: 27

Given:

```
class A {  
    public A(){  
        System.out.print("A ");  
    }  
}  
  
class B extends A{  
    public B(){  
        System.out.print("B "); //line n1  
    }  
}  
  
class C extends B{  
  
    public C(){ //line n2  
        System.out.print("C ");  
    }  
    public static void main(String[] args) {  
        C c = new C();  
    }  
}
```

What is the result?

**A.**  
C B A

**B.**  
C

**C.**  
A B C

**D.**  
Compilation fails at line n1 and line n2

**Answer:** C

**Explanation:**

## QUESTION NO: 28

Given:

```
class X {  
    static int i;  
    int j;  
    public static void main(String[] args) {  
        X x1 = new X();  
        X x2 = new X();  
        x1.i = 3;  
        x1.j = 4;  
        x2.i = 5;  
        x2.j = 6;  
        System.out.println(  
            x1.i + " " +  
            x1.j + " " +  
            x2.i + " " +  
            x2.j);  
    }  
}
```

What is the result?

**A.**

3 4 5 6

**B.**

3 4 3 6

**C.**

5 4 5 6

**D.**

3 6 4 6

**Answer: C**

**Explanation:**

#### QUESTION NO: 29

Given the code fragment:

```

1. public class Test {
2.     public static void main(String[] args) {
3.         /* insert code here */
4.         array[0]=10;
5.         array[1]=20;
6.         System.out.print(array[0]+":"+array[1]);
7.     }
8. }

```

Which code fragment, when inserted at line 3, enables the code to print 10:20?

**A.**

int[] array n= new int[2];

**B.**

int[] array;

array = int[2];

**C.**

int array = new int[2];

**D.**

int array [2] ;

**Answer: B**

**Explanation:**

### QUESTION NO: 30

Given the code fragment:

```

public static void main(String[] args) {
    String[] arr = {"A", "B", "C", "D"};
    for (int i = 0; i < arr.length; i++) {
        System.out.print(arr[i] + " ");
        if (arr[i].equals("C")) {
            continue;
        }
        System.out.println("Work done");
        break;
    }
}

```

What is the result?

- A.**  
A B C Work done
- B.**  
A B C D Work done
- C.**  
A Work done
- D.**  
Compilation fails

**Answer: C**

**Explanation:**

#### **QUESTION NO: 31**

Which three are advantages of the Java exception mechanism?

- A.**  
Improves the program structure because the error handling code is separated from the normal program function
- B.**  
Provides a set of standard exceptions that covers all the possible errors
- C.**  
Improves the program structure because the programmer can choose where to handle exceptions
- D.**  
Improves the program structure because exceptions must be handled in the method in which they occurred
- E.**  
Allows the creation of new exceptions that are tailored to the particular program being created

**Answer: A,C,D**

Reference:<http://javajee.com/introduction-to-exceptions-in-java>

#### **QUESTION NO: 32**

Given the code from the Greeting.Java file:

```
public class Greeting {  
    public static void main(String[] args) {  
        System.out.println("Hello " + args[0]);  
    }  
}
```

Which set of commands prints Hello Duke in the console?

- A) javac Greeting  
java Greeting Duke
- B) javac Greeting.java Duke  
java Greeting
- C) javac Greeting.java  
java Greeting Duke
- D) javac Greeting.java  
java Greeting.class Duke

A.

Option A

B.

Option B

C.

Option C

D.

Option D

**Answer: C**

**Explanation:**

**QUESTION NO: 33**

Given:

```

class Alpha {
    int ns;
    static int s;
    Alpha(int ns) {
        if (s < ns) {
            s = ns;
            this.ns = ns;
        }
    }
    void doPrint() {
        System.out.println("ns = " + ns + " s = " + s);
    }
}

```

And,

```

public class TestA {
    public static void main(String[] args) {
        Alpha ref1 = new Alpha(50);
        Alpha ref2 = new Alpha(125);
        Alpha ref3 = new Alpha(100);
        ref1.doPrint();
        ref2.doPrint();
        ref3.doPrint();
    }
}

```

What is the result?

- A) ns = 50 s = 125  
ns = 125 s = 125  
ns = 100 s = 125
- B) ns = 50 s = 125  
ns = 125 s = 125  
ns = 0 s = 125
- C) ns = 50 s = 50  
ns = 125 s = 125  
ns = 100 s = 100
- D) ns = 50 s = 50  
ns = 125 s = 125  
ns = 0 s = 125

A.

Option A

B.

Option B

C.

Option C

D.

Option D

**Answer: B**

**Explanation:**

**QUESTION NO: 34**

Given the code fragment:

```
public static void main(String[] args) {  
    int ii = 0;  
    int jj = 7;  
    for (ii = 0; ii < jj - 1; ii = ii + 2) {  
        System.out.print(ii + " ");  
    }  
}
```

What is the result?

A.

2 4

B.

0 2 4 6

C.

0 2 4

D.

Compilation fails

**Answer: C**

**Explanation:**

**QUESTION NO: 35**

Given the code fragment:

```
LocalDate date1 = LocalDate.now();
LocalDate date2 = LocalDate.of(2014, 6, 20);
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);
System.out.println("date1 = " + date1);
System.out.println("date2 = " + date2);
System.out.println("date3 = " + date3);
```

Assume that the system date is June 20, 2014. What is the result?

- A) date1 = 2014-06-20  
date2 = 2014-06-20  
date3 = 2014-06-20
- B) date1 = 06/20/2014  
date2 = 2014-06-20  
date3 = Jun 20, 2014
- C) Compilation fails.
- D) A DateParseException is thrown at runtime.

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**Answer: D**

**Explanation:**

### QUESTION NO: 36

Given the code fragment:

```
7. StringBuilder sb1 = new StringBuilder("Duke");
8. String str1 = sb1.toString();
9. // insert code here
10. System.out.print(str1 == str2);
```

Which code fragment, when inserted at line 9, enables the code to print true?

**A.**

String str2 = str1;

**B.**

String str2 = new String (str1);

**C.**

String str2 = sb1. toString ();

**D.**

String str2 = "Duke";

**Answer: B**

**Explanation:**

### QUESTION NO: 37

Given the code fragment:

```
public class Test {  
    static int count = 0;  
    int i = 0;  
  
    public void changeCount() {  
        while (i < 5) {  
            i++;  
            count++;  
        }  
    }  
  
    public static void main(String[] args) {  
        Test check1 = new Test();  
        Test check2 = new Test();  
        check1.changeCount();  
        check2.changeCount();  
        System.out.print(check1.count + " : " + check2.count);  
    }  
}
```

What is the result?

**A.**

10 : 10

**B.**

5 : 5

**C.**

5 : 10

**D.**

Compilation fails

**Answer: A****Explanation:****QUESTION NO: 38**

Given the code fragment:

```
public static void main(String[] args) {
    double discount = 0;
    int qty = Integer.parseInt(args[0]);
    //line n1;
}
```

And given the requirements:

If the value of the qty variable is greater than or equal to 90, discount = 0.5

If the value of the qty variable is between 80 and 90, discount = 0.2

Which two code fragments can be independently placed at line n1 to meet the requirements?

- A) if (qty >= 90) { discount = 0.5; }
 if (qty > 80 && qty < 90) { discount = 0.2; }
- B) discount = (qty >= 90) ? 0.5 : 0;
 discount = (qty > 80) ? 0.2 : 0;
- C) discount = (qty >= 90) ? 0.5 : (qty > 80)? 0.2 : 0;
- D) if (qty > 80 && qty < 90) {
 discount = 0.2;
 } else {
 discount = 0;
 }
 if (qty >= 90) {
 discount = 0.5;
 } else {
 discount = 0;
 }
- E) discount = (qty > 80) ? 0.2 : (qty >= 90) ? 0.5 : 0;

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**E.**

Option E

**Answer: A,C**

**Explanation:**

### QUESTION NO: 39

Given:

```
public class Test {  
  
    public static void main(String[] args) {  
        if (args[0].equals("Hello") ? false : true) {  
            System.out.println("Success");  
        } else {  
            System.out.println("Failure");  
        }  
    }  
}
```

And given the commands:

javac Test.java

Java Test Hello

What is the result?

**A.**

Success

**B.**

Failure

**C.**

Compilation fails.

**D.**

An exception is thrown at runtime

**Answer: B**

**Explanation:**

#### **QUESTION NO: 40**

Which three statements describe the object-oriented features of the Java language?

**A.**

Objects cannot be reused.

**B.**

A subclass can inherit from a superclass.

**C.**

Objects can share behaviors with other objects.

**D.**

A package must contain more than one class.

**E.**

Object is the root class of all other objects.

**F.**

A main method must be declared in every class.

**Answer: B,C,F**

Reference:<http://www.javaworld.com/article/2075459/java-platform/java-101--object-oriented-language-basics--part-5--object-and-its-methods.html>(see the sub title, Object is root of all classes not all other objects)

#### **QUESTION NO: 41**

Given the following code:

```
public static void main(String[] args){  
    String[] planets = {"Mercury", "Venus", "Earth", "Mars"};  
  
    System.out.println(planets.length);  
    System.out.println(planets[1].length());  
}
```

What is the output?

**A.**

4

4

**B.**

3

5

**C.**

4

7

**D.**

5

4

**E.**

4

5

**F.**

4

21

**Answer: E**

**Explanation:**

#### QUESTION NO: 42

You are developing a banking module. You have developed a class named ccMask that has a maskcc method.

Given the code fragment:

```
class CCmask {
    public static String maskCC(String creditCard) {
        String x = "XXXX-XXXX-XXXX-";
        //line n1
    }

    public static void main(String[] args) {
        System.out.println(maskCC("1234-5678-9101-1121"));
    }
}
```

You must ensure that the maskCC method returns a string that hides all digits of the credit card number except the four last digits (and the hyphens that separate each group of four digits).

Which two code fragments should you use at line n1, independently, to achieve this requirement?

- A) `StringBuilder sb = new StringBuilder(creditCard);  
sb.substring(15, 19);  
return x + sb;`
- B) `return x + creditCard.substring(15, 19);`
- C) `StringBuilder sb = new StringBuilder(x);  
sb.append(creditCard, 15, 19);  
return sb.toString();`
- D) `StringBuilder sb = new StringBuilder(creditCard);  
StringBuilder s = sb.insert(0, x);  
return s.toString();`

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**Answer: B,C**

**Explanation:**

**QUESTION NO: 43**

Given:

Acc.java:

```
package p1;
public class Acc {
    int p;
    private int q;
    protected int r;
    public int s;
}
```

Test.java:

```
package p2;
import p1.Acc;
public class Test extends Acc {
    public static void main(String[] args) {
        Acc obj = new Test();
    }
}
```

Which statement is true?

**A.**

Both p and s are accessible by obj.

**B.**

Only s is accessible by obj.

**C.**

Both r and s are accessible by obj.

**D.**

p, r, and s are accessible by obj.

**Answer: B**

**Explanation:**

#### QUESTION NO: 44

Given:

Base.java:

```
class Base {  
    public void test(){  
        System.out.println("Base ");  
    }  
}
```

DerivedA.java:

```
class DerivedA extends Base {  
    public void test(){  
        System.out.println("DerivedA ");  
    }  
}
```

DerivedB.java:

```
class DerivedB extends DerivedA {  
    public void test(){  
        System.out.println("DerivedB ");  
    }  
    public static void main(String[] args) {  
        Base b1 = new DerivedB();  
        Base b2 = new DerivedA();  
        Base b3 = new DerivedB();  
        b1 = (Base) b3;  
        Base b4 = (DerivedA) b3;  
        b1.test();  
        b4.test();  
    }  
}
```

What is the result?

**A.**

Base

DerivedA

**B.**

Base

DerivedB

**C.**

DerivedB

DerivedB

D.

DerivedB

DerivedA

E.

A classcast Exception is thrown at runtime.

**Answer: C**

**Explanation:**

#### QUESTION NO: 45

Given the code fragment:

```
public static void main(String[] args) {
    ArrayList myList = new ArrayList();
    String[] myArray;
    try {
        while (true) {
            myList.add("My String");
        }
    }
    catch (RuntimeException re) {
        System.out.println("Caught a RuntimeException");
    }
    catch (Exception e) {
        System.out.println("Caught an Exception");
    }
    System.out.println("Ready to use");
}
```

What is the result?

A.

Execution terminates in the first catch statement, and caught a RuntimeException is printed to the console.

B.

Execution terminates in the second catch statement, and caught an Exception is printed to the console.

C.

A runtime error is thrown in the thread "main".

D.

Execution completes normally, and Ready to use is printed to the console.

**E.**

The code fails to compile because a throws keyword is required.

**Answer: C**

**Explanation:**

**QUESTION NO: 46**

Given:

```
System.out.println("5 + 2 = " + 3 + 4);  
System.out.println("5 + 2 = " + (3 + 4));
```

What is the result?

- A) 5 + 2 = 34  
5 + 2 = 34
- B) 5 + 2 + 3 + 4  
5 + 2 = 7
- C) 7 = 7  
7 + 7
- D) 5 + 2 = 34  
5 + 2 = 7

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**Answer: D**

**Explanation:****QUESTION NO: 47**

Given the code fragments:

Person.java:

```
public class Person {
    String name;
    int age;

    public Person(String n, int a) {
        name = n;
        age = a;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {
    for (Person p : list) {
        if (predicate.test(p)) {
            System.out.println(p.name + " ");
        }
    }
}

public static void main(String[] args) {
    List<Person> iList = Arrays.asList(new Person("Hank", 45),
                                         new Person("Charlie", 40),
                                         new Person("Smith", 38));
    //line n1
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

**A.**

checkAge (iList, ( ) -> p. get Age ( ) > 40);

**B.**

checkAge(iList, Person p -> p.getAge( ) > 40);

**C.**

checkAge (iList, p -> p.getAge ( ) > 40);

**D.**

checkAge(iList, (Person p) -> { p.getAge() > 40; });

**Answer: C**

**Explanation:****QUESTION NO: 48**

Given the code fragment:

```
public static void main(String[] args) {  
    String[][] arr = {{ "A", "B", "C"}, {"D", "E"}};  
    for (int i = 0; i < arr.length; i++) {  
        for (int j = 0; j < arr[i].length; j++) {  
            System.out.print(arr[i][j] + " ");  
            if (arr[i][j].equals("B")) {  
                break;  
            }  
        }  
        continue;  
    }  
}
```

What is the result?

**A.**

A B C

**B.**

A B C D E

**C.**

A B D E

**D.**

Compilation fails.

**Answer: C**

**Explanation:****QUESTION NO: 49**

Given the code fragment:

```
public static void main(String[] args) {  
    String str = " ";  
    str.trim();  
    System.out.println(str.equals("") + " " + str.isEmpty());  
}
```

What is the result?

A.

true true

B.

true false

C.

false false

D.

false true

**Answer: C**

**Explanation:**

## QUESTION NO: 50

Given the code fragment:

```
public class App {  
    public static void main(String[] args) {  
        String str1 = "Java";  
        String str2 = new String("java");  
        //line n1  
        {  
            System.out.println("Equal");  
        } else {  
            System.out.println("Not Equal");  
        }  
    }  
}
```

Which code fragment, when inserted at line n1, enables the App class to print Equal?

- A) String str3 = str2;  
    if (str1 == str3)
- B) if (str1.equalsIgnoreCase(str2))
- C) String str3 = str2;  
    if (str1.equals(str3))
- D) if (str1.toLowerCase() == str2.toLowerCase())

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**Answer: B**

**Explanation:**

### QUESTION NO: 51

Given:

```
public class SumTest {  
  
    public static void doSum(Integer x, Integer y) {  
        System.out.println("Integer sum is " + (x + y));  
    }  
  
    public static void doSum(double x, double y) {  
        System.out.println("double sum is " + (x + y));  
    }  
  
    public static void doSum(float x, float y) {  
        System.out.println("float sum is " + (x + y));  
    }  
  
    public static void doSum(int x, int y) {  
        System.out.println("int sum is " + (x + y));  
    }  
  
    public static void main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

What is the result?

- A) int sum is 30  
float sum is 30.0
- B) int sum is 30  
double sum is 30
- C) Integer sum is 30  
double sum is 30.0
- D) Integer sum is 30  
float sum is 30.0

A.

Option A

B.

Option B

C.

Option C

D.

Option D

**Answer: B**

**Explanation:**

**QUESTION NO: 52**

Given the code fragment:

```
String[] strs = new String[2];
int idx = 0;
for (String s : strs) {
    strs[idx].concat(" element " + idx);
    idx++;
}
for (idx = 0; idx < strs.length; idx++) {
    System.out.println(strs[idx]);
}
```

What is the result?

**A.**

Element 0

Element 1

**B.**

Null element 0

Null element 1

**C.**

Null

Null

**D.**

A NullPointerException is thrown at runtime.

**Answer: D**

**Explanation:**

**QUESTION NO: 53**

Given:

```
class Vehicle {  
    int x;  
    Vehicle(){  
        this(10); // line n1  
    }  
    Vehicle(int x) {  
        this.x = x;  
    }  
}  
  
class Car extends Vehicle {  
    int y;  
    Car() {  
        super();  
        this(20); // line n2  
    }  
    Car(int y) {  
        this.y = y;  
    }  
    public String toString() {  
        return super.x + ":" + this.y;  
    }  
}
```

And given the code fragment:

And given the code fragment:

```
Vehicle y = new Car();  
System.out.println(y);
```

What is the result?

A.

10:20

B.

0:20

C.

Compilation fails at line n1

**D.**

Compilation fails at line n2

**Answer: D****Explanation:****QUESTION NO: 54**

Given the definitions of the MyString class and the Test class:

MyString.java:

```
package p1;
class MyString {
    String msg;
    MyString(String msg) {
        this.msg = msg;
    }
}
```

Test.java:

```
package p1;
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello " + new StringBuilder("Java SE 8"));
        System.out.println("Hello " + new MyString("Java SE 8"));
    }
}
```

What is the result?

- A) Hello Java SE 8  
Hello Java SE 8
- B) Hello java.lang.StringBuilder@<<hashcode1>>  
Hello p1.MyString@<<hashcode2>>
- C) Hello Java SE 8  
Hello p1.MyString@<<hashcode>>
- D) Compilation fails at the Test class.

**A.**

Option A

**B.**

Option B

**C.**

Option C

D.

Option D

**Answer: C**

**Explanation:**

**QUESTION NO: 55**

Given the code fragment:

```
3. public static void main(String[] args) {  
4.     int iVar = 100;  
5.     float fVar = 100.100f;  
6.     double dVar = 123;  
7.     iVar = fVar;  
8.     fVar = iVar;  
9.     dVar = fVar;  
10.    fVar = dVar;  
11.    dVar = iVar;  
12.    iVar = dVar;  
13. }
```

Which three lines fail to compile?

A.

Line 7

B.

Line 8

C.

Line 9

D.

Line 10

E.

Line 11

F.

Line 12

**Answer: A,D,F**

**Explanation:****QUESTION NO: 56**

Given:

MainTest.java:

```
public class MainTest {  
  
    public static void main(int[] args) {  
        System.out.println("int main " + args[0]);  
    }  
    public static void main(Object[] args) {  
        System.out.println("Object main " + args[0]);  
    }  
    public static void main(String[] args) {  
        System.out.println("String main " + args[0]);  
    }  
}
```

and commands:

```
javac MainTest.java  
java MainTest 1 2 3
```

What is the result?

**A.**

int main 1

**B.**

Object main 1

**C.**

String main 1

**D.**

Compilation fails

**E.**

An exception is thrown at runtime

**Answer: C**

**Explanation:**

**QUESTION NO: 57**

Given the code fragment:

```
int num[][] = new int[1][3];
for (int i = 0; i < num.length; i++) {
    for (int j = 0; j < num[i].length; j++) {
        num[i][j] = 10;
    }
}
```

Which option represents the state of the num array after successful completion of the outer loop?

- A) num[0][0]=10  
num[0][1]=10  
num[0][2]=10
- B) num[0][0]=10  
num[1][0]=10  
num[2][0]=10
- C) num[0][0]=10  
num[0][1]=0  
num[0][2]=0
- D) num[0][0]=10  
num[0][1]=10  
num[0][2]=10  
num[0][3]=10  
num[1][0]=0  
num[1][1]=0  
num[1][2]=0  
num[1][3]=0

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**Answer: A****Explanation:****QUESTION NO: 58**

Given the code fragment:

```
public class Person {  
    String name;  
    int age = 25;  
  
    public Person(String name) {  
        this(); //line n1  
        setName(name);  
    }  
  
    public Person(String name, int age) {  
        Person(name); //line n2  
        setAge(age);  
    }  
  
    //setter and getter methods go here  
  
    public String show() {  
        return name + " " + age + " " + number ;  
    }  
    public static void main(String[] args) {  
        Person p1 = new Person("Jesse");  
        Person p2 = new Person("Walter",52);  
        System.out.println(p1.show());  
        System.out.println(p2.show());  
    }  
}
```

What is the result?

**A.**

Jesse 25

Walter 52

**B.**

Compilation fails only at line n1

**C.**

Compilation fails only at line n2

**D.**

Compilation fails at both line n1 and line n2

**Answer: B**

**Explanation:**

**QUESTION NO: 59**

Given the following code for a Planet object:

```
public class Planet {  
    public String name;  
    public int moons;  
  
    public Planet(String name, int moons) {  
        this.name = name;  
        this.moons = moons;  
    }  
}
```

And the following main method:

```
public static void main(String[] args) {  
    Planet[] planets = {  
        new Planet("Mercury", 0),  
        new Planet("Venus", 0),  
        new Planet("Earth", 1),  
        new Planet("Mars", 2)  
    };  
  
    System.out.println(planets);  
    System.out.println(planets[2]);  
    System.out.println(planets[2].moons);  
}
```

What is the output?

- A) planets  
Earth  
1
- B) [LPlanets.Planet;@15db9742  
Earth  
1
- C) [LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
1
- D) [LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
[LPlanets.Moon;@7852e922
- E) [LPlanets.Planet;@15db9742  
Venus  
0

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**E.**

Option E

**Answer: C**

**Explanation:**

#### **QUESTION NO: 60**

You are asked to develop a program for a shopping application, and you are given the following information:

The application must contain the classes Toy, EduToy, and ConsToy. The Toy class is the superclass of the other two classes.

The int calculatePrice (Toy t) method calculates the price of a toy.

The void printToy (Toy t) method prints the details of a toy.

Which definition of the Toy class adds a valid layer of abstraction to the class hierarchy?

- A) public abstract class Toy{  
    public abstract int calculatePrice(Toy t);  
    public void printToy(Toy t) { /\* code goes here \*/ }  
}
- B) public abstract class Toy {  
    public int calculatePrice(Toy t) ;  
    public void printToy(Toy t) ;  
}
- C) public abstract class Toy {  
    public int calculatePrice(Toy t) ;  
    public final void printToy(Toy t){ /\* code goes here \*/ }  
}
- D) public abstract class Toy {  
    public abstract int calculatePrice(Toy t) { /\* code goes here \*/ }  
    public abstract void printToy(Toy t) { /\* code goes here \*/ }  
}

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**Answer: A**

**Explanation:**

## QUESTION NO: 61

Given the following code:

```
int[] intArr = {15, 30, 45, 60, 75};  
intArr[2] = intArr[4];  
intArr[4] = 90;
```

What are the values of each element in intArr after this code has executed?

**A.**

15, 60, 45, 90, 75

**B.**

15, 90, 45, 90, 75

**C.**

15, 30, 75, 60, 90

**D.**

15, 30, 90, 60, 90

**E.**

15, 4, 45, 60, 90

**Answer: C**

**Explanation:**

## QUESTION NO: 62

Given the following array:

```
int[] intArr = {8, 16, 32, 64, 128};
```

Which two code fragments, independently, print each element in this array?

- A) 

```
for (int i : intArr) {
    System.out.print(intArr[i] + " ");
}
```
- B) 

```
for (int i : intArr) {
    System.out.print(i + " ");
}
```
- C) 

```
for (int i=0 : intArr) {
    System.out.print(intArr[i] + " ");
    i++;
}
```
- D) 

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(i + " ");
}
```
- E) 

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(intArr[i] + " ");
}
```
- F) 

```
for (int i; i < intArr.length; i++) {
    System.out.print(intArr[i] + " ");
}
```

A.

Option A

B.

Option B

C.

Option C

D.

Option D

E.

Option E

F.

Option F

**Answer: B,E**

**Explanation:**

**QUESTION NO: 63**

Given the content of three files:

A.java:

```
public class A {  
    public void a() {}  
    int a;  
}
```

B.java:

```
public class B {  
    private int doStuff() {  
        private int x = 100;  
        return x++;  
    }  
}
```

C.java:

```
import java.io.*;  
package p1;  
class A {  
    public void main(String fileName) throws IOException {}  
}
```

Which statement is true?

**A.**

Only the A.Java file compiles successfully.

**B.**

Only the B.java file compiles successfully.

**C.**

Only the C.java file compiles successfully.

**D.**

The A.Java and B.java files compile successfully.

**E.**

The B.java and C.java files compile successfully.

**F.**

The A.Java and C.java files compile successfully.

**Answer: A**

**Explanation:**

**QUESTION NO: 64**

Given the code fragment:

```
int[] array = {1, 2, 3, 4, 5};
```

And given the requirements:

1. Process all the elements of the array in the order of entry.
2. Process all the elements of the array in the reverse order of entry.
3. Process alternating elements of the array in the order of entry.

Which two statements are true?

**A.**

Requirements 1, 2, and 3 can be implemented by using the enhanced for loop.

**B.**

Requirements 1, 2, and 3 can be implemented by using the standard for loop.

**C.**

Requirements 2 and 3 CANNOT be implemented by using the standard for loop.

**D.**

Requirement 1 can be implemented by using the enhanced for loop.

**E.**

Requirement 3 CANNOT be implemented by using either the enhanced for loop or the standard for loop.

**Answer: D,E**

**Explanation:**

**QUESTION NO: 65**

Given:

```
public class TestScope {  
    public static void main(String[] args) {  
        int var1 = 200;  
        System.out.print(doCalc(var1));  
        System.out.print(" "+var1);  
    }  
    static int doCalc(int var1){  
        var1 = var1 * 2;  
        return var1;  
    }  
}
```

What is the result?

**A.**  
400 200

**B.**  
200 200

**C.**  
400 400

**D.**  
Compilation fails.

**Answer: A**

**Explanation:**

#### QUESTION NO: 66

Given the following class declarations:

```
public abstract class Animal  
  
public interface Hunter  
  
public class Cat extends Animal implements Hunter  
  
public class Tiger extends Cat
```

Which answer fails to compile?

- A) `ArrayList<Animal> myList = new ArrayList<>();  
myList.add(new Tiger());`
- B) `ArrayList<Hunter> myList = new ArrayList<>();  
myList.add(new Cat());`
- C) `ArrayList<Hunter> myList = new ArrayList<>();  
myList.add(new Tiger());`
- D) `ArrayList<Tiger> myList = new ArrayList<>();  
myList.add(new Cat());`
- E) `ArrayList<Animal> myList = new ArrayList<>();  
myList.add(new Cat());`

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**E.**

Option E

**Answer: E**

**Explanation:**

#### **QUESTION NO: 67**

Which statement is true about Java byte code?

**A.**

It can run on any platform.

**B.**

It can run on any platform only if it was compiled for that platform.

**C.**

It can run on any platform that has the Java Runtime Environment.

**D.**

It can run on any platform that has a Java compiler.

**E.**

It can run on any platform only if that platform has both the Java Runtime Environment and a Java compiler.

**Answer: D**

Reference:<http://www.math.uni-hamburg.de/doc/java/tutorial/getStarted/intro/definition.html>

Explanation:

Java bytecodes help make "write once, run anywhere" possible. You can compile your program into bytecodes on any platform that has a Java compiler. The bytecodes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

## QUESTION NO: 68

Given:

```
public class MarkList {  
    int num;  
    public static void graceMarks(MarkList obj4) {  
        obj4.num += 10;  
    }  
    public static void main(String[] args) {  
        MarkList obj1 = new MarkList();  
        MarkList obj2 = obj1;  
        MarkList obj3 = null;  
        obj2.num = 60;  
        graceMarks(obj2);  
    }  
}
```

How many MarkList instances are created in memory at runtime?

**A.**

1

**B.**

2

**C.**

3

D.

4

**Answer: A**

**Explanation:**

**QUESTION NO: 69**

Given:

```
public class Triangle {  
    static double area;  
    int b = 2, h = 3;  
    public static void main(String[] args) {  
        double p, b, h; //line n1  
        if (area == 0){  
            b = 3;  
            h = 4;  
            p = 0.5;  
        }  
        area = p * b * h; //line n2  
        System.out.println("Area is " + area);  
    }  
}
```

What is the result?

**A.**

Area is 6.0

**B.**

Area is 3.0

**C.**

Compilation fails at line n1

**D.**

Compilation fails at line n2.

**Answer: D**

**Explanation:**

**QUESTION NO: 70**

Given the code fragment:

```
public class Test {  
    public static void main(String[] args) {  
        //line n1  
        switch (x) {  
            case 1:  
                System.out.println("One");  
                break;  
            case 2:  
                System.out.println("Two");  
                break;  
        }  
    }  
}
```

Which three code fragments can be independently inserted at line n1 to enable the code to print one?

**A.**

Byte x = 1;

**B.**

short x = 1;

**C.**

String x = "1";

**D.**

Long x = 1;

**E.**

Double x = 1;

**F.**

Integer x = new Integer ("1");

**Answer: A,B,F**

**Explanation:**

**QUESTION NO: 71**

Given:

```
public class App {  
  
    public static void main(String[] args) {  
        Boolean[] bool = new Boolean[2];  
  
        bool[0] = new Boolean(Boolean.parseBoolean("true"));  
        bool[1] = new Boolean(null);  
  
        System.out.println(bool[0] + " " + bool[1]);  
    }  
}
```

What is the result?

**A.**

True false

**B.**

True null

**C.**

Compilation fails

**D.**

A NullPointerException is thrown at runtime

**Answer: A**

**Explanation:**

## QUESTION NO: 72

Given the following code for the classes MyException and Test:

```
public class MyException extends RuntimeException {}  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            method1();  
        }  
        catch (MyException ne) {  
            System.out.print("A");  
        }  
    }  
    public static void method1() { // line n1  
        try {  
            throw Math.random() > 0.5 ? new MyException() : new RuntimeException();  
        }  
        catch (RuntimeException re) {  
            System.out.print("B");  
        }  
    }  
}
```

What is the result?

- A.  
A
- B.  
B
- C.  
Either A or B
- D.  
A B
- E.  
A compile time error occurs at line n1

**Answer: B**

**Explanation:**

### QUESTION NO: 73

Given:

```
public class App {  
  
    String myStr = "7007";  
  
    public void doStuff(String str) {  
        int myNum = 0;  
        try {  
            String myStr = str;  
            myNum = Integer.parseInt(myStr);  
        } catch (NumberFormatException ne) {  
            System.out.println("Error");  
        }  
        System.out.println(  
            "myStr: " + myStr + ", myNum: " + myNum);  
    }  
  
    public static void main(String[] args) {  
        App obj = new App();  
        obj.doStuff("9009");  
    }  
}
```

What is the result?

- A.  
myStr: 9009, myNum: 9009

**B.**

myStr: 7007, myNum: 7007

**C.**

myStr: 7007, myNum: 9009

**D.**

Compilation fails

**Answer: C**

**Explanation:**

#### **QUESTION NO: 74**

Which two are benefits of polymorphism?

**A.**

Faster code at runtime

**B.**

More efficient code at runtime

**C.**

More dynamic code at runtime

**D.**

More flexible and reusable code

**E.**

Code that is protected from extension by other classes

**Answer: B,D**

Reference:<https://www.cs.princeton.edu/courses/archive/fall98/cs441/mainus/node5.html>

#### **QUESTION NO: 75**

Given the code fragment:

```
int nums1[] = new int[3];
int nums2[] = {1, 2, 3, 4, 5};
nums1 = nums2;
for (int x : nums1) {
    System.out.print(x + ":");
}
```

What is the result?

**A.**

1:2:3:4:5:

**B.**

1:2:3:

**C.**

Compilation fails.

**D.**

An ArrayoutofBoundsException is thrown at runtime.

**Answer: A**

**Explanation:**

#### QUESTION NO: 76

Given:

```
public class Product {  
    int id;  
    String name;  
    public Product(int id, String name) {  
        this.id = id;  
        this.name = name;  
    }  
}
```

And given the code fragment:

```
4. Product p1 = new Product(101, "Pen");  
5. Product p2 = new Product(101, "Pen");  
6. Product p3 = p1;  
7. boolean ans1 = p1 == p2;  
8. boolean ans2 = p1.name.equals(p2.name);  
9. System.out.print(ans1 + ":" + ans2);
```

What is the result?

A.

true:true

B.

true:false

C.

false:true

D.

false:false

**Answer: C**

**Explanation:**

**QUESTION NO: 77**

Given the following classes:

```
public class Employee {  
    public int salary;  
}  
  
public class Manager extends Employee {  
    public int budget;  
}  
  
public class Director extends Manager {  
    public int stockOptions;  
}
```

And given the following main method:

```
public static void main(String[] args) {  
    Employee employee = new Employee();  
    Manager manager = new Manager();  
    Director director = new Director();  
    //line n1  
}
```

Which two options fail to compile when placed at line n1 of the main method?

**A.**

employee.salary = 50\_000;

**B.**

director.salary = 80\_000;

**C.**

employee.budget = 200\_000;

**D.**

manager.budget = 1\_000\_000;

**E.**

manager.stockOption = 500;

**F.**

director.stockOptions = 1\_000;

**Answer: C,E**

**Explanation:**

**QUESTION NO: 78**

Which one of the following code examples uses valid Java syntax?

A.

```
public class Boat {  
  
    public static void main (String [] args) {  
        System.out.println ("I float.");  
    }  
}
```

B.

```
public class Cake {  
    public static void main (String [] ) {  
        System.out.println ("Chocolate");  
    }  
}
```

C.

```
public class Dog {  
    public void main (String [] args) {  
        System.out.println ("Squirrel.");  
    }  
}
```

D.

```
public class Bank {  
    public static void main (String () args) {  
        System.out.println ("Earn interest.");  
    }  
}
```

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**Answer: A**

Reference:<https://docs.oracle.com/javase/tutorial/getStarted/application/>

**QUESTION NO: 79**

Given the code fragment:

```
int n [] [] = {{1, 3}, {2, 4}};
for (int i = n.length-1; i >= 0; i--) {
    for (int y : n[i]) {
        System.out.print (y);
    }
}
```

What is the result?

**A.**  
1324

**B.**  
2413

**C.**  
3142

**D.**  
4231

**Answer: C**

**Explanation:**

**QUESTION NO: 80**

Given:

```
class Caller {  
    private void init () {  
        System.out.println("Initialized");  
    }  
  
    private void start () {  
        init();  
        System.out.println("Started");  
    }  
}  
  
public class TestCall {  
    public static void main(String[] args) {  
        Caller c = new Caller();  
        c.start();  
        c.init();  
    }  
}
```

What is the result?

**A.**

An exception is thrown at runtime.

**B.**

Initialized

Started

Initialized

**C.**

Initialized

Started

**D.**

Compilation fails.

**Answer: D**

**Explanation:**

**QUESTION NO: 81**

Given the code fragment:

```
public static void main(String[] args) {  
    try {  
        int num = 10;  
        int div = 0;  
        int ans = num / div;  
    } catch (ArithmaticException ae) {  
        ans = 0 // line n1  
    } catch (Exception e) {  
        System.out.println("Invalid calculation");  
    }  
    System.out.println("Answer = " + ans); // line n2  
}
```

What is the result?

**A.**

Answer = 0

**B.**

Invalid calculation

**C.**

Compilation fails only atline n1.

**D.**

Compilation fails only atline n2.

**E.**

Compilation fails only atline n1andline2.

**Answer: E**

**Explanation:**

## QUESTION NO: 82

Given:

```
public class MyField {  
    int x;  
    int y;  
    public void doStuff(int x, int y) {  
        this.x = x;  
        y = this.y;  
    }  
    public void display () {  
        System.out.print(x + " " + y + " : ");  
    }  
    public static void main(String[] args) {  
        MyField m1 = new MyField();  
        m1.x = 100;  
        m1.y = 200;  
        MyField m2 = new MyField();  
        m2.doStuff(m1.x, m1.y);  
        m1.display();  
        m2.display();  
    }  
}
```

What is the result?

**A.**

100 0 : 100 200:

**B.**

100 0 : 100 0 :

**C.**

100 200 : 100 200 :

**D.**

100 200 : 100 0 :

**Answer: B**

**Explanation:**

### QUESTION NO: 83

Given:

```
public class Vowel {  
    private char var;  
    public static void main(String[] args) {  
        char var1 = 'a';  
        char var2 = var1;  
        var2 = 'e';  
  
        Vowel obj1 = new Vowel ();  
        Vowel obj2 = obj1;  
        obj1.var = 'i';  
        obj2.var = 'o';  
  
        System.out.println(var1 + ", " +var2);  
        System.out.print(obj1.var + ", " +obj2.var);  
    }  
}
```

What is the result?

**A.**

e, e

i, o

**B.**

a, e

i, o

**C.**

a,e

o, o

**D.**

e, e

o, o

**Answer: A**

**Explanation:**

**QUESTION NO: 84**

Given the code fragment:

```
if (aVar++ < 10) {  
    System.out.println(aVar + " Hello World!");  
} else {  
    System.out.println(aVar + " Hello Universe!");  
}
```

What is the result if the integer aVar is 9?

**A.**

Compilation fails.

**B.**

10 Hello Universe!

**C.**

10 Hello World!

**D.**

9 Hello World!

**Answer: C**

**Explanation:**

### QUESTION NO: 85

Given:

```
public class MyClass {  
    public static void main(String[] args) {  
        String s = "Java Duke";  
        int len = s.trim().length();  
        System.out.print(len);  
    }  
}
```

What is the result?

**A.**

Compilation fails.

**B.**

11

C.

8

D.

9

E.

10

**Answer: D**

**Explanation:**

### QUESTION NO: 86

Given:

```
public class Test {  
    public static void main(String[] args) {  
        boolean a = new Boolean(Boolean.valueOf(args[0]));  
        boolean b = new Boolean(args[1]);  
        System.out.println(a + " " + b);  
    }  
}
```

And given the commands:

javac Test.java

java Test TRUE null

What is the result?

A.

TRUE null

B.

true false

C.

false false

D.

true true

**E.**

A `ClassCastException` is thrown at runtime.

**Answer: D****Explanation:****QUESTION NO: 87**

Given the code fragments:

**A.java:**

```
package p1;
public class A { }
```

**B.java:**

```
package p1.p2;
//line n1
public class B {
    public void doStuff() {
        A b = new A();
    }
}
```

**C.java:**

```
package p3;
//line n2
public class C {
    public static void main(String[] args) {
        A o1 = new A();
        B o2 = new B();
    }
}
```

Which modification enables the code to compile?

**A.**

Replace line n1 with:

import p1.A;

Replace line n2 with:

import p1.A;

import p1.p2.B;

**B.**

Replace line n1 with:

import p1;

Replace line n2 with:

import p1;

import p1.p2;

**C.**

Replace line n1 with:

import p1.A;

Replace line n2 with:

import p1.\*;

**D.**

Replace line n1 with:

import p1.\*;

Replace line n2 with:

import p1.p2.\*;

**Answer: D**

**Explanation:**

## QUESTION NO: 88

Which statement will empty the contents of a StringBuilder variable named sb?

**A.**

sb.deleteAll();

**B.**

sb.delete(0, sb.size());

**C.**

sb.delete(0, sb.length());

**D.**

sb.removeAll();

**Answer: C**

**Explanation:**

### QUESTION NO: 89 CORRECT TEXT

Given:

```
String stuff = "TV";
String res = null;

if (stuff.equals ("TV")) {
res = "Walter";
} else if (stuff.equals ("Movie") ) {
res= "White";
} else {
res= "No Result";
}
```

Which code fragment can replace the if block?

- A) stuff.equals ("TV") ? res= "Walter" : stuff.equals ("Movie") ? res = "White" : res = "No Result";
- B) res = stuff.equals ("TV") ? "Walter" else stuff.equals ("Movie")? "White" : "No Result";
- C) res = stuff.equals ("TV") ? stuff.equals ("Movie")? "Walter" : "White" : "No Result";
- D) res = stuff.equals ("TV")? "Walter" : stuff.equals ("Movie")? "White" : "No Result";

Answer:

**B**

**QUESTION NO: 90**

Given:

```
class Patient {  
    String name;  
    public Patient (String name) {  
        this.name = name;  
    }  
}
```

And the code fragment:

```
8. public class Test {  
9.     public static void main (String [] args) {  
10.         List ps = new ArrayList ();  
11.         Patient p2 = new Patient ("Mike");  
12.         ps.add(p2);  
13.  
14.         // insert code here  
15.  
16.         if (f >= 0) {  
17.             System.out.print ("Mike Found");  
18.         }  
19.     }  
20. }
```

Which code fragment, when inserted at line 14, enables the code to print Mike Found?

**A.**

```
int f = ps.indexOf (p2)
```

**B.**

```
int f = ps.indexOf (Patient ("Mike"));
```

**C.**

```
int f = ps.indexOf (new Patient "Mike");
```

**D.**

Patient p = new Patient ("Mike");

Int f = ps.indexOf (p)

**Answer: A**

**Explanation:**

**QUESTION NO: 91**

Which statement is true about the switch statement?

**A.**

It must contain the default section.

**B.**

The break statement, at the end of each case block, is mandatory.

**C.**

Its case label literals can be changed at runtime.

**D.**

Its expression must evaluate to a single value.

**Answer: D**

Reference:<http://www.dummies.com/programming/java/switch-statements-in-java/>

**QUESTION NO: 92**

Given:

```

class Animal {
    String type = "Canine";
    int maxSpeed = 60;

    Animal () {}

    Animal (String type, int maxSpeed) {
        this.type = type;
        this.maxSpeed = maxSpeed;
    }
}

class WildAnimal extends Animal {
    String bounds;

    WildAnimal (String bounds) {
        //line n1
    }

    WildAnimal (String type, int maxSpeed,
                //line n2
    }

}

```

And given the code fragment:

```

7. WildAnimal wolf = new WildAnimal ("Long");
8. WildAnimal tiger = new WildAnimal ("Feline", 80, "Short");
9. System.out.println (wolf.type + " " + wolf.maxSpeed + " " +
wolf.bounds);
10. System.out.println (tiger.type + " " + tiger.maxSpeed + " " +
tiger.bounds);

```

Which two modifications enable the code to print the following output?

Canine 60 Long

Feline 80 Short

**A.**

Replace line n1 with:

super ();

this.bounds = bounds;

**B.**

Replace line n1 with:

this.bounds = bounds;

super ();

**C.**

Replace line n2 with:

super (type, maxSpeed);

this (bounds);

**D.**

Replace line n1 with:

this ("Canine", 60);

this.bounds = bounds

**E.**

Replace line n2 with:

super (type, maxSpeed);

this.bounds = bounds;

**Answer: A**

**Explanation:**

### QUESTION NO: 93

Given the code fragment:

```
public static void main (String [] args) {  
    String names [] = {"Thomas", "Peter", "Joseph");  
    String pws [] = new String [3];  
    int idx = 0;  
    try {  
        for (String n: names) {  
            pws [idx] = n.substring (2, 6);  
            idx++;  
        }  
    }  
    catch (Exception e) {  
        System.out.println ("Invalid Name");  
    }  
    for (String p: pws) {  
        System.out.println (p);  
    }  
}
```

What is the result?

**A.**

Invalid Name

**B.**

Invalid Name

omas

**C.**

Invalid Name

omas

null

null

**D.**

omas

ter

seph

**Answer: C**

**Explanation:**

**QUESTION NO: 94**

Given the code fragment:

```

class Employee {
    private String name;
    private int age;
    private int salary;

    public Employee (String name, int age) {
        setName (name)
        setAge (age)
        setSalary (2000);
    }
    public Employee (String name, int age, int salary) {
        setSalary (salary);
        this (name, age);
    }
    //getter and setter methods for attributes go here
    public void printDetails () {
        System.out.println (name + " : " + age + " : " + salary);
    }
}

```

Test.java

```

class Test {
    public static void main (String [] args {
        Employee e1 = new Employee ();
        Employee e2 = new Employee ("Jack, 50");
        Employee e3 = new Employee ("Chloe", 40, 5000);
        e1.printDetails ();
        e2.printDetails ();
        e3.printDetails ();
    }
}

```

Which is the result?

**A.**

Compilation fails in the Employee class.

**B.**

null : 0: 0

Jack : 50 : 0

Chloe : 40 : 5000

**C.**

null : 0 : 0

Jack : 50 : 2000

Chloe : 40 : 5000

**D.**

Compilation fails in the Test class.

**E.**

Both the Employee class and the test class fail to compile.

**Answer: E**

**Explanation:**

### QUESTION NO: 95

Given the code fragments:

A.java:

```
package p1;
public class A {  
}
```

B.java:

```
package p1.p2;  
//line n1
public class B {
    public void doStuff () {
        A b = new A ();
    }
}
```

C.java

```
package p3;
//line n2
public class C {
    public static void main (String [] args) {
        A 01 = new A ();
        B 02 = new B ();
    }
}
```

Which modification enables the code to compile?

**A.**

Replace line n1 with:

```
import p1.*;
```

Replace line n2 with:

```
import p1. p2.*;
```

**B.**

Replace line n1 with:

```
import p1. A;
```

Replace line n2 with:

```
import p1.*;
```

**C.**

Replace line n1 with:

```
import p1. A;
```

Replace line n2 with:

```
import p1. A;
```

```
import p1. p2.B ;
```

**D.**

Replace line n1 with:

```
import p1;
```

Replace line n2 with:

```
import p1;
```

```
import p1. p2;
```

**Answer: C**

**Explanation:**

## QUESTION NO: 96

Given:

```
class A {  
    public void test () {  
        System.out.println ("A");  
    }  
}  
class B extends A {  
    public void test () {  
        System.out.println ("B");  
    }  
}  
public class C extends A {  
    public void test () {  
        System.out.println ("C");  
    }  
  
    public static void main (String [] args) {  
        A b1 = new A ();  
        A b2 = new C ();  
        b1 = (A) b2;  
        A b3 = (B) b2;           //line n1  
        A b3 = (B) b2;           //line n2  
        b1.test ();  
        b3.test ();  
    }  
}
```

What is the result?

**A.**

A

B

**B.**

A

C

**C.**

C

C

**D.**

A ClassCastException is thrown only at line n1.

**E.**

A ClassCastException is thrown only at line n2.

**Answer: B****Explanation:****QUESTION NO: 97****Given:**

```
public class SumTest {  
  
    public static void doSum(Integer x, Integer y) {  
        System.out.println("Integer sum is " + (x + y));  
    }  
  
    public static void doSum(double x, double y) {  
        System.out.println("double sum is " + (x + y));  
    }  
  
    public static void doSum(float x, float y) {  
        System.out.println("float sum is " + (x + y));  
    }  
  
    public static void doSum(int x, int y) {  
        System.out.println("int sum is " + (x + y));  
    }  
  
    public static void main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

What is the result?

**A.**

int sum is 30

float sum is 30.0

**B.**

int sum is 30

double sum is30.0

**C.**

integer sum is 30

double sum is 30.0

D.

integer sum is 30

float sum is 30.0

**Answer: D**

**Explanation:**

### QUESTION NO: 98

Given the code fragment:

```
4. class X {  
5.     public void printFileContent () {  
6.         /* code goes here */  
7.         throw new IOException ();  
8.     }  
9.}  
10. public class Test {.  
11.     public static void main (String [] args) {  
12.         X xobj = new X ();  
13.         xobj.printFileContent ();  
14.     }  
15. }
```

Which two modifications should you make so that the code compiles successfully?

- A. At line 14, insert `throw new IOException ();`
- B. Replace line 5 with `public void printFileContent () throws IOException {`
- C. Replace line 11 with `public static void main (String [] args) throws Exception {`
- D. Replace line 13 with:

```
try {  
    xobj.printFileContent ();  
}  
catch (Exception e) {}  
catch (IOException e) {}
```

- E. Replace line 7 with `throw IOException ("Exception raised");`

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**E.**

Option E

**Answer: E**

**Explanation:**

### QUESTION NO: 99

You are asked to create a method that accepts an array of integers and returns the highest value from that array.

Given the code fragment:

```
class Test {  
    public static void main (String [] args) {  
        int numbers [] = {12, 13, 42, 32, 15, 156, 23, 51, 12};  
        int max = findMax (numbers);  
    }  
/*line n1 */ {  
    int max = 0;  
    /* code goes here*/  
    return max;  
}  
}
```

Which method signature do you use at line n1?

**A.**

public int findMax (int [] numbers)

**B.**

static int[] findMax (int max)

**C.**

static int findMax (int [] numbers)

**D.**

```
final int findMax (int [] )
```

**Answer: A**

**Explanation:**

**QUESTION NO: 100**

Which three statements are true about the structure of a Java class?

**A.**

A public class must have a main method.

**B.**

A class can have only one private constructor.

**C.**

A method can have the same name as a field.

**D.**

A class can have overloaded static methods.

**E.**

The methods are mandatory components of a class.

**F.**

The fields need not be initialized before use.

**Answer: A,C,E**

**Explanation:**

**QUESTION NO: 101**

Given the code fragment:

```
Public static void main (String[] args) {  
    System.out.println ("Result A " + 0 + 1);  
    System.out.println ("Result B " + (1) + (2) );  
}
```

What is the result?

A. Result A 1

Result B 3

B. Result A 01

Result B 3

C. Result A 01

Result B 12

D. Result A 1

Result B 12

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**Answer: C**

**Explanation:**

## QUESTION NO: 102

Given:

```
public class App {  
    int count;  
    public static void displayMsg () {  
        count++;  
        System.out.println ("Welcome "+"Visit Count: "+count); // line n1  
    }  
    public static void main (String [] args) {  
        App.displayMsg ();  
        App.displayMsg ();  
    }  
}
```

What is the result?

**A.**

Compilation fails at line n3 and line n4.

**B.**

Compilation fails at line n1 and line n2.

**C.**

Welcome Visit Count:1

Welcome Visit Count: 2

**D.**

Welcome Visit Count:1

Welcome Visit Count: 2

**Answer: B**

**Explanation:**

### QUESTION NO: 103

Given the code fragment:

```
public class Person {  
    String name;  
    int age = 25;  
  
    public Person (String name) {  
        this (); // //line n1  
        setName (name);  
    }  
    public Person (String name, int age) {  
        Person (name); //line n2  
        setAge (age);  
    }  
    //setter and getter methods go here  
  
    public String show () {  
        return name + " " + age;  
    }  
    public static void main (String [] args) {  
        Person p1 = new Person ("Jesse");  
        Person p2 = new Person ("Walter", 52);  
        System.out.println (p1.show () );  
        System.out.println (p2.show () );  
    }  
}
```

What is the result?

**A.**

Compilation fails at both line n1 and line n2.

**B.**

Compilation fails only at line n2.

**C.**

Compilation fails only at line n1.

**D.**

Jesse 25

Walter 52

**Answer: D**

**Explanation:**

**QUESTION NO: 104**

Given the code fragment:

```
public class Test {  
  
    static int count = 0  
    int i = 0;  
  
    public void changeCount () {  
        while (i<5) {  
            i++;  
            count++;  
        }  
    }  
  
    public static void main (String [] args) {  
        Test check1 = new Test ();  
        Test check2 = new Test ();  
        check1.changeCount ();  
        check2.changeCount ();  
        System.out. print (check1.count + " : " + check2.count);  
    }  
}
```

What is the result?

A.

5 : 5

B.

10 : 10

C.

5 : 10

D.

Compilation fails.

**Answer: B**

Reference:

## QUESTION NO: 105

Given the code fragment:

```
public static void main (String [] args) {  
    ArrayList<Integer> points = new ArrayList<> ();  
    points.add (1);  
    points.add (2);  
    points.add (3);  
    points.add (4);  
    points.add (null);  
    points.remove (2);  
    points.remove (null);  
    System.out.println(points);  
}
```

What is the result?

**A.**

A NullPointerException is thrown at runtime.

**B.**

[1, 2, 4]

**C.**

[1, 2, 4, null ]

**D.**

[1, 3, 4, null ]

**E.**

[1, 3, 4 ]

**F.**

Compilation fails.

**Answer: F**

**Explanation:**

Your Code ...

```

1 public static void main (String [] args) {
2     ArrayList<Integer> points = new ArrayList<> () ;
3     points.add (1) ;
4     points.add (2) ;
5     points.add (3) ;
6     points.add (4) ;
7     points.add (null) ;
8     points.remove (null) ;
9     System.out.printIn (points) ;
10 }
```

External Libraries ...

[+ Add External Library \(from Maven Repo\)](#)

cs1.keyboard

Input Arguments (args of Main Method)...

Interactive mode :  OFF

Stdin Inputs...

[Execute](#)[Save](#)[My Projects](#)[Recent](#)[Collaborate](#)[Others](#) ▾[Goto Another Language/DB](#) ▾

Result...

compiled and executed in 0 second(s)

No "public class" found to execute

**QUESTION NO: 106**

Given:

```

class Test {
    public static void main (String [] args) {
        int numbers [ ] ;
        numbers = new int [2] ;
        numbers [0] = 10 ;
        numbers [1] = 20 ;

        numbers = new int [4] ;
        numbers [2] = 30 ;
        numbers [3] = 40 ;
        for (int x : numbers) {
            System.out.print (" " + x) ;
        }
    }
}
```

What is the result?

**A.**

10 20 30 40

**B.**

0 0 30 40

**C.**

Compilation fails.

**D.**

An exception is thrown at runtime.

**Answer: C**

**Explanation:**

#### QUESTION NO: 107

Which code fragment causes a compilation error?

- A. float flt = 100F;
- B. float flt = (float) 1\_11.00;
- C. float flt = 100;
- D. double y1 = 203.22;  
    float flt = y1;
- E. int y2 = 100;  
    float flt = (float) y2;

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**E.**

Option E

**Answer: D****Explanation:****QUESTION NO: 108**

Given:

```
public class Fieldinit {  
    char c;  
    boolean b;  
    float f;  
    void printAll() {  
        System.out.println ("c = " + c);  
        System.out.println ("b = " + b);  
        System.out.println ("f = " + f);  
    }  
    public static void main (String [] args) {  
        FieldInit f = new FieldInit ();  
        f.printAll ();  
    }  
}
```

What is the result?

**A.**

c=

b = false

f = 0.0

**B.**

c= null

b = true

f = 0.0

**C.**

c=0

b = false

f = 0.0f

**D.**

c= null

b = false

f = 0.0F

**Answer: C**

**Explanation:**

### QUESTION NO: 109

Which three statements are true about exception handling?

**A.**

Only unchecked exceptions can be rethrown.

**B.**

All subclasses of the RuntimeException class are recoverable.

**C.**

The parameter in a catch block is of Throwable type.

**D.**

All subclasses of the RuntimeException class must be caught or declared to be thrown.

**E.**

All subclasses of the Exception class except the RuntimeException class are checked exceptions.

**F.**

All subclasses of the Error class are checked exceptions and are recoverable.

**Answer: C,E,F**

**Explanation:**

### QUESTION NO: 110

Given the code fragment:

```
public static void main (String [ ] args) {  
    int [] stack = {10,20,30}  
    int size = 3;  
    int idx = 0;  
    /*line n1 */  
    System.out.print ("The Top element: " + stack [idx] );  
}
```

Which code fragment, inserted at line n1, prints The Top element: 30?

- A. do {  
 idx++;  
 } while (idx >= size);
- B. while (idx < size) {  
 idx++;  
 }
- C. do {  
 idx++;  
 } while (idx < size -1);
- D. do {  
 idx++;  
 } while (idx<= size);
- E. while (idx <= size -1) {  
 idx++  
 }

**A.**

Option A

**B.**

Option B

**C.**

Option C

D.

Option D

E.

Option E

**Answer: A**

**Explanation:**

**QUESTION NO: 111**

Given the code fragment:

```
public static void main (String [] args) {  
    String myStr = "Hello World";  
    myStr.trim ();  
    int il = myStr.indexOf (" ");  
    System.out.println (il);  
}
```

What is the result?

A.

An exception is thrown at runtime.

B.

-1

C.

5

D.

0

**Answer: A**

**Explanation:**

**QUESTION NO: 112**

Given:

```
class Equal {  
    public static void main (String [] args) {  
        String str1 = "Java";  
        String [] str2 = { "J", "a", "v", "a"};  
        String str3 = "";  
        for (String str : str2) {  
            str3 = str3+str;  
        }  
        boolean b1 = (str1== str3);  
        boolean b2 = (str1.equals (str3));  
        System.out.print (b1+", "+b2);  
    }  
}
```

What is the result?

- A.**  
false, false
- B.**  
false, true
- C.**  
true, false
- D.**  
true, true

**Answer: B**

**Explanation:**

### QUESTION NO: 113

Which two statements are true?

- A.**  
Error class is unextendable.
- B.**  
Error class is extendable.
- C.**  
Error is a RuntimeException.

**D.**

Error is an Exception.

**E.**

Error is a Throwable.

**Answer: B,C**

**Explanation:**

#### QUESTION NO: 114

Given the code fragment:

```
public static void main (String[ ] args) {  
    int data [ ] = {2010, 2013, 2014, 2015, 2014};  
    int key = 2014;  
    int count = 0;  
    for (int e: data) {  
        if (e! = key) {  
            continue;  
            count++;  
        }  
    }  
    System.out.print (count + "Found");  
}
```

What is the result?

**A.**

Compilation fails.

**B.**

0 Found

**C.**

1 Found

**D.**

3 Found

**Answer: D**

**Explanation:**

**QUESTION NO: 115**

Given the code fragment:

```
LocalDate Time dt= LocalDateTime.of(2014, 7, 31, 1, 1);
dt.plusDays (30);
dt. plusMonths (1);
System.out.print (dt format (DateTimeFormatter. ISO_DATE) );
```

What is the result?

**A.**

An exception is thrown at runtime.

**B.**

07-31-2014

**C.**

2014-07-31

**D.**

2014-09-30

**Answer: C**

**Explanation:**

**QUESTION NO: 116**

Given:

```
public class Test {  
    public static final int MIN =1;  
    public static void main (String [] args) {  
        int x = args.length;  
        if (checkLimit (x)) { //line n1  
            System.out.println ("Java SE");  
        } else {  
            System.out.println ("Java EE");  
        }  
    }  
    public static boolean checkLimit (int x) {  
        return (x > = MIN) ? true : false;  
    }  
}
```

And given the commands:

```
javac Test.java
```

```
java Test
```

What is the result?

- A.**  
Java SE
- B.**  
Java EE
- C.**  
Compilation fails at line n1.
- D.**  
A NullPointerException is thrown at runtime.

**Answer: A**

**Explanation:**

### QUESTION NO: 117

Given:

```
interface Readable {  
    public void readBook();  
    public void setBookMark();  
}  
  
abstract class Book implements Readable { // line n1  
    public void readBook() {}  
    // line n2  
}  
  
class EBook extends Book { // line n3  
    public void readBook() {}  
    // line n4  
}
```

And given the code fragment:

```
Book book1 = new EBook();
```

```
Book1.readBook();
```

Which option enables the code to compile?

- A. Replace the code fragment at line n3 with:  
abstract class EBook extends Book {
- B. Replace the code fragment at line n1 with:  
class Book implements Readable {
- C. At line n2 insert:  
public abstract void setBookMark ();
- D. At line n4 insert:  
public void setBookMark () {}

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**Answer: A**

**Explanation:**

**QUESTION NO: 118**

Given the following class:

```
public class CheckingAccount {  
    public int amount:  
        // line n1  
}
```

And given the following main method, located in another class:

```
public static void main (String [] args) {  
    CheckingAccount acct = new CheckingAccount ();  
    //line n2  
}
```

Which three pieces of code, when inserted independently, set the value of amount to 100?

- A. At line n2 insert:  
amount = 100;
- B. At line n2 insert:  
This. amount = 100
- C. At line n2 insert:  
acct.amount = 100
- D. At line n1 insert:  
public CheckingAccount () {  
 amount = 100;  
}
- E. At line n1 insert:  
public CheckingAccount () {  
 this.amount = 100;  
}
- F. At line n1 insert:  
public CheckingAccount () {  
 acct.amount = 100;  
}

**A.**

Option A

**B.**

Option B

**C.**

Option C

**D.**

Option D

**E.**

Option E

**F.**

Option F

**Answer: B,C,E**

**Explanation:****QUESTION NO: 119**

Given the code fragments:

```
Interface Exportable {  
    Void export();  
}  
  
class Tool implements Exportable {  
    protected void export () {          //line n1  
        System.out.println("Tool::export");  
    }  
}  
  
class ReportTool extends Tool implements Exportable {  
  
    public void export() {             //line n2  
        System.out.println("RTool::export");  
    }  
  
    public static void main(String[] args) {  
        Tool aTool = new ReportTool();  
        Tool bTool = new Tool();  
        callExport(aTool);  
        callExport(bTool);  
    }  
  
    public static void callExport (Exportable ex) {  
        ex.export();  
    }  
}
```

What is the result?

**A.**

Compilation fails only at line n2.

**B.**

RTool::export

Tool::export

**C.**

Tool::export

Tool:export

D.

Compilation fails only at line n1.

E.

Compilation fails at both line n1 and line n2.

**Answer: E**

**Explanation:**

### QUESTION NO: 120

Given the code fragment:

```
24. float var1 = (12_345.01 >= 123_45.00) ? 12_456 : 124_56.02f;  
25. float var2 = var1 + 1024;  
26. System.out.print(var2);
```

What is the result?

A.

An exception is thrown at runtime.

B.

Compilation fails.

C.

13480.0

D.

13480.02

**Answer: C**

**Explanation:**

### QUESTION NO: 121

Given:

```
public class Test {  
    public static int stVar = 100;  
    public int var = 200;  
    public String toString() {  
        return var + ":" + stVar;  
    }  
}
```

And given the code fragment:

```
Test t1 = new Test();  
t1.var = 300;  
System.out.println(t1);  
Test t2 = new Test();  
t2.stVar = 300;  
System.out.println(t2);
```

What is the result?

**A.**

300:300

200:300

**B.**

300:100

200:300

**C.**

300:0

0:300

**D.**

200:300

200:300

**Answer: D**

**Explanation:**

**QUESTION NO: 122**

Given:

```
class C2 {  
    public void displayC2() {  
        System.out.print("C2");  
    }  
}  
interface I {  
    public void displayI();  
}  
class C1 extends C2 implements I {  
    public void displayI() {  
        System.out.print("C1");  
    }  
}
```

And given the code fragment:

```
C2 obj1 = new C1();  
I obj2 = new C1();  
  
C2 s = obj2;  
I t = obj1;  
  
t.displayI();  
s.displayC2()
```

What is the result?

- A.**  
C2C2
- B.**  
C1C2
- C.**  
C1C1
- D.**  
Compilation fails

**Answer: A**

**Explanation:****QUESTION NO: 123**

Given:

```
package clothing;
public class Shirt {
    public static String getColor() {
        return "Green";
    }
}
```

Given the code fragment:

```
package clothing.pants;
// line n1
public class Jeans {
    public void matchShirt(){
        //line n2
        if(color.equals("Green")) {
            System.out.print("Fit")
        }
    }
    public static void main (String[] args) {
        Jeans trouser = new Jeans();
        trouser.matchShirt();
    }
}
```

Which two sets of actions, independently, enable the code fragment to print Fit?

**A.**

At line n1 insert:import clothing.Shirt;

At line n2 insert:String color = getColor();

**B.**

At line n1 insert:import clothing.\*;

At line n2 insert:String color = Shirt.getColor();

C.

At line n1 insert:import static clothing.Shirt.getColor;

At line n2 insert:String color = getColor();

D.

At line n1 no changes required.

At line n2 insert:String color = Shirt.getColor();

E.

At line n1 insert:import clothing;

At line n2 insert:String color = Shirt.getColor();

**Answer: A**

**Explanation:**

#### QUESTION NO: 124

Given the code fragments:

```
class Student {  
    String name;  
    int age;  
}
```

And,

```
4. public class Test {  
5.     public static void main(String[] args) {  
6.         Student s1 = new Student();  
7.         Student s2 = new Student();  
8.         Student s3 = new Student();  
9.         s1 = s3;  
10.        s3 = s2;  
11.        s2 = null;  
12.    }  
13.}
```

Which statement is true?

**A.**

After line 11, three objects are eligible for garbage collection.

**B.**

After line 11, two objects are eligible for garbage collection.

**C.**

After line 11, one object is eligible for garbage collection.

**D.**

After line 11, none of the objects are eligible for garbage collection.

**Answer: C****Explanation:****QUESTION NO: 125**

Given the code fragment:

```
int wd = 0;
String days[] = {"sun", "mon", "wed", "sat"};
for (String s:days) {
    switch (s) {
        case "sat":
        case "sun":
            wd -= 1;
            break;
        case "mon":
            wd++;
        case "wed":
            wd += 2;
    }
}
System.out.println(wd);
```

What is the result?

**A.**

3

**B.**

4

C.

-1

D.

Compilation fails.

**Answer: B**

**Explanation:**

### QUESTION NO: 126

Given the code fragment:

```
public static void main(String[] args) {  
    LocalDate date = LocalDate.of(2012, 01, 32);  
    date.plusDays(10);  
    System.out.println(date);  
}
```

What is the result?

A.

2012-02-10

B.

2012-02-11

C.

Compilation fails

D.

A DateTimeException is thrown at runtime.

**Answer: C**

**Explanation:**

### QUESTION NO: 127

Given:

```

public class App {
    public static void main(String[] args) {
        int i = 10;
        int j = 20;
        int k = j += i / 5;
        System.out.print(i + " : " + j + " : " + k);
    }
}

```

What is the result?

**A.**

10 : 30 : 6

**B.**

10 : 22 : 22

**C.**

10 : 22 : 20

**D.**

10 : 22 : 6

**Answer: B**

**Explanation:**

Explanation

Your Code ...

```

1- public class App {
2-     public static void main (String[] args) {
3-         int i = 10;
4-         int j = 20;
5-         int k = j += i / 5;
6-         System.out.print (i + " : " + j + " : " + k);
7-     }
8- }
9

```

External Libraries ...

CommandLine Arguments ...

Interactive mode :  OFF Version: JDK 9.0.1

Stdin Inputs...

Result...

CPU Time: 0.20 sec(s), Memory: 32080 kilobyte(s) compiled and executed in 1.229 sec(s)

10 : 22 : 22

**QUESTION NO: 128**

Given:

```
interface Downloadable {  
    public void download();  
}  
  
interface Readable extends Downloadable {      // line n1  
    public void readBook();  
}  
  
abstract class Book implements Readable {        // line n2  
    public void readBook() {  
        System.out.println("Read Book");  
    }  
}  
  
class EBook extends Book {                      // line n3  
    public void readBook() {  
        System.out.println("Read E-Book");  
    }  
}
```

And given the code fragment:

```
Book book1 = new EBook();  
book1.readBook();
```

What is the result?

- A.**  
Compilation fails at line n2.
- B.**  
Read Book
- C.**  
Read E-Book
- D.**  
Compilation fails at line n1.
- E.**  
Compilation fails at line n3.

**Answer: B**

**Explanation:**

**QUESTION NO: 129**

Given the following class:

```
public class Rectangle {  
    private double length;  
    private double height;  
    private double area;  
  
    public void setLength(double length) {  
        this.length = length;  
    }  
    public void setHeight(double height) {  
        this.height = height;  
    }  
    public void setArea() {  
        area = length*height;  
    }  
}
```

Which two changes would encapsulate this class and ensure that the area field is always equal to length \* height whenever the Rectangle class is used?

**A.**

Call the setArea method at the end of the setHeight method.

**B.**

Call the setArea method at the beginning of the setHeight method.

**C.**

Call the setArea method at the end of the setLength method.

**D.**

Call the setArea method at the beginning of the setLength method.

**E.**

Change the setArea method to private.

**F.**

Change the area field to public.

**Answer: A,E**

**Explanation:**

**QUESTION NO: 130**

Given the code fragment:

```
13. List colors = new ArrayList();
14. colors.add("green");
15. colors.add("red");
16. colors.add("blue");
17. colors.add("yellow");
18. colors.remove(2);
19. colors.add(3, "cyan");
20. System.out.print(colors);
```

What is the result?

**A.**

(green, red, yellow, cyan)

**B.**

(green, blue, yellow, cyan)

**C.**

(green, red, cyan, yellow)

**D.**

AnIndexOutOfBoundsException is thrown at runtime.

**Answer: C**

**Explanation:**

**QUESTION NO: 131**

Given the code fragment:

```
abstract class Toy {
    int price;
    // line n1
}
```

Which three code fragments are valid at line n1?

**A.**

```
public static void insertToy() {  
    /* code goes here */  
}
```

**B.**

```
public abstract Toy getToy() {  
  
    return new Toy();  
}
```

**C.**

```
public void printToy();
```

**D.**

```
public int calculatePrice() {  
  
    return price;  
}
```

**E.**

```
public abstract int computeDiscount();
```

**Answer: C,D,E**

**Explanation:**

**QUESTION NO: 132**

Given:

```
public class Test {  
    int x, y;  
  
    public Test(int x, int y) {  
        initialize(x, y);  
    }  
  
    public void initialize(int x, int y) {  
        this.x = x * x;  
        this.y = y * y;  
    }  
  
    public static void main(String[] args) {  
        int x = 3, y = 5;  
        Test obj = new Test(x, y);  
        System.out.println(x + " " + y);  
    }  
}
```

What is the result?

**A.**

Compilation fails.

**B.**

3 5

**C.**

0 0

**D.**

9 25

**Answer: B**

**Explanation:**

**QUESTION NO: 133**

Given the code fragment:

```
public static void main(String[] args) {  
    int array[] = {10, 20, 30, 40, 50};  
    int x = array.length;  
    /* line n1 */  
}
```

Which two code fragments can be independently inserted at line n1 to enable the code to print the elements of the array in reverse order?

**A.**

```
while (x > 0) {  
  
    x--;  
  
    System.out.print(array[x]);  
  
}
```

**B.**

```
do {  
  
    x--;  
  
    System.out.print(array[x]);  
  
} while (x >= 0);
```

**C.**

```
while (x >= 0) {  
  
    System.out.print(array[x]);  
  
}
```

**D.**

```
do {  
  
    System.out.print(array[x]);  
  
}
```

```
--x;  
  
} while (x >= 0);
```

**E.**

```
while (x > 0) {  
  
    System.out.print(array[--x]);  
  
}
```

**Answer: B,E****Explanation:****QUESTION NO: 134**

Given:

```
class Test  
  
int a1;  
  
public static void doProduct(int a) {  
    a = a * a;  
}  
  
public static void doString(StringBuilder s) {  
    s.append(" " + s);  
}  
  
public static void main(String[] args) {  
    Test item = new Test();  
    item.a1 = 11;  
    StringBuilder sb = new StringBuilder("Hello");  
    Integer i = 10;  
    doProduct(i);  
    doString(sb);  
    doProduct(item.a1);  
    System.out.println(i + " " + sb + " " + item.a1);  
}
```

What is the result?

**A.**

10 Hello Hello 11

**B.**

10 Hello Hello 121

**C.**

100 Hello 121

**D.**

100 Hello Hello 121

**E.**

10 Hello 11

**Answer: B**

**Explanation:**

### QUESTION NO: 135

Given the code fragment:

```
public static void main (String[] args) {  
    String[] arr = {"Hi", "How", "Are", "You"};  
    List<String> arrList = new ArrayList<>(Arrays.asList(arr));  
    if (arrList.removeIf((String s) -> (return s.length() <= 2;))) {  
        System.out.println(s + "removed")  
    }  
}
```

What is the result?

**A.**

Compilation fails.

**B.**

Hi removed

**C.**

An UnsupportedOperationException is thrown at runtime.

**D.**

The program compiles, but it prints nothing.

**Answer: A**

**Explanation:**

**QUESTION NO: 136**

Which two class definitions fail to compile?

**A.**

```
abstract class A3 {  
    private static int i;  
    public void doStuff(){  
    public A3(){  
    }  
}
```

**B.**

```
final class A1 {  
    public A1(){  
    }  
}
```

**C.**

```
public class A2 {  
    private static int i;  
    private A2(){  
    }  
}
```

**D.**

```
class A4 {  
    protected static final int i;  
    private void doStuff(){  
    }  
}
```

**E.**

```
final abstract class A5 {  
    protected static int i;  
}
```

```
void doStuff(){}
abstract void doIt();
}
```

**Answer: C,E**

**Explanation:**

### QUESTION NO: 137

Given:

```
class Student {
    String name;
    public Student(String name) {
        this.name = name;
    }
}

public class Test {
    public static void main(String[] args) {
        Student[] students = new Student[3];
        students[1] = new Student("Richard");
        students[2] = new Student("Donald");
        for (Student s : students) {
            System.out.println(" " + s.name);
        }
    }
}
```

What is the result?

**A.**

null

Richard

Donald

**B.**

Richard

Donald

**C.**

Compilation fails.

**D.**

An `ArrayIndexOutOfBoundsException` is thrown at runtime.

**E.**

A `NullPointerException` is thrown at runtime.

**Answer:** A

**Explanation:**

### QUESTION NO: 138

The following grid shows the state of a 2D array:

0	0	
	X	0
	X	X

This grid is created with the following code:

```
char[][] grid = new char[3][3];
grid[1][1] = 'X';
grid[0][0] = '0';
grid[2][1] = 'X';
grid[0][1] = '0';
grid[2][2] = 'X';
grid[1][2] = '0';
```

Which line of code, when inserted in place of //line n1, adds an X into the grid so that the grid contains three consecutive X's?

**A.**

`grid[1][3] = 'X';`

**B.**

`grid[3][1] = 'X';`

**C.**

grid[0][2] = 'X';

**D.**

grid[2][0] = 'X';

**E.**

grid[1][2] = 'X';

**Answer: C**

**Explanation:**

### QUESTION NO: 139

Given:

```
public class Test {
    public static void main(String[] args) {
        int x = 1;
        int y = 0;
        if(x++ > ++y) {
            System.out.print("Hello ");
        } else {
            System.out.print("Welcome ");
        }
        System.out.print("Log " + x + ":" + y);
    }
}
```

What is the result?

**A.**

Hello Log 1:0

**B.**

Hello Log 2:1

**C.**

Welcome Log 2:1

**D.**

Welcome Log 1:0

**Answer: C**

**Explanation:****QUESTION NO: 140**

Given the code snippet from a compiled Java source file:

```
public class MyFile
{
    public static void main (String[] args)
    {
        String arg1 = args[1];
        String arg2 = args[2];
        String arg3 = args[3];
        System.out.println("Arg is " + arg3);
    }
}
```

Which command-line arguments should you pass to the program to obtain the following output?

Arg is 2

**A.**

java MyFile 1 3 2 2

**B.**

java MyFile 2 2 2

**C.**

java MyFile 1 2 2 3 4

**D.**

java MyFile 0 1 2 3

**Answer: A**

**Explanation:****QUESTION NO: 141**

Given the code fragment:

```
public static void main(String[] args) {  
    int[] arr = {1, 2, 3, 4};  
    int i = 0;  
    do {  
        System.out.print(arr[i] + " ");  
        i++;  
    } while (i < arr.length - 1);  
}
```

What is the result?

**A.**

1 2 3 4

followed by an `ArrayIndexOutOfBoundsException`

**B.**

1 2 3

**C.**

1 2 3 4

**D.**

Compilation fails.

**Answer: A**

**Explanation:**

## QUESTION NO: 142

Given:

```
public class Test {  
    public static void main(String[] args) {  
        Test ts = new Test();  
        System.out.print(isAvailable + " ");  
        isAvailable= ts.doStuff();  
        System.out.println(isAvailable);  
    }  
    public static boolean doStuff() {  
        return !isAvailable;  
    }  
    static boolean isAvailable = false;  
}
```

What is the result?

**A.**

Compilation fails.

**B.**

false true

**C.**

true false

**D.**

true true

**E.**

false false

**Answer: B**

**Explanation:**

➤ **Vendor: Oracle**

➤ **Exam Code: 1Z0-808**

➤ **Exam Name: Java SE 8 Programmer I**

➤ **New Questions (Mar/2017)**

[Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps](#)

**NEW QUESTION 121**

Given:

```
public class TestLoop {  
    public static void main(String[] args) {  
        int array[] = {0, 1, 2, 3, 4};  
        int key = 3;  
        for (int pos = 0; pos < array.length; ++pos) {  
            if (array[pos] == key) {  
                break;  
            }  
        }  
        System.out.print("Found " + key + " at " + pos);  
    }  
}
```

What is the result?

- A. Found 3 at 2
- B. Found 3 at 3
- C. Compilation fails
- D. An exception is thrown at runtime

**Answer: C**

**Explanation:**

The following line does not compile:

```
System.out.print("Found " + key + " at " + pos);
```

The variable pos is undefined at this line, as its scope is only valid in the for loop. Any variables created inside of a loop are LOCAL TO THE LOOP.

**NEW QUESTION 122**

Given:

```
import java.util.*;
public class Ref {
    public static void main(String[] args) {
        StringBuilder s1 = new StringBuilder("Hello Java!");
        String s2 = s1.toString();
        List<String> lst = new ArrayList<String>();
        lst.add(s2);
        System.out.println(s1.getClass());
        System.out.println(s2.getClass());
        System.out.println(lst.getClass());
    }
}
```

What is the result?

- A. class java.lang.String  
class java.lang.String  
class java.util.ArrayList
- B. class java.lang.Object  
class java.lang.Object  
class java.util.Collection
- C. class java.lang.StringBuilder  
class java.lang.String  
class java.util.ArrayList
- D. class java.lang.StringBuilder  
class java.lang.String  
class java.util.List

**Answer: C**

**Explanation:**

```
class java.lang.StringBuilder
class java.lang.String
class java.util.ArrayList
```

### NEW QUESTION 123

Given:

```
public class Case {
    public static void main(String[] args) {
        String product = "Pen";
        product.toLowerCase();
        product.concat(" Box".toLowerCase());
        System.out.print(product.substring(4, 6));
    }
}
```

What is the result?

- A. box
- B. nbo
- C. bo
- D. nb
- E. An exception is thrown at runtime

**Answer: E**

### NEW QUESTION 124

Given:

```
1. public class Whizlabs {  
2.     public static void main(String[] args) {  
3.         int sum = 0;  
4.         for(int x = 0; x <= 10; x++)  
5.             sum += x;  
6.         System.out.print("Sum for 0 to " + x);  
7.         System.out.println(" = " + sum);  
8.     }  
9. }  
10. }
```

Which is true?

- A. Sum for 0 to 0 = 55
- B. Sum for 0 to 10 = 55
- C. Compilation fails due to error on line 6
- D. Compilation fails due to error on line 7
- E. An Exception is thrown at the runtime

**Answer: D**

**Explanation:**

Loop variables scope limited to that enclosing loop. So in this case, the scope of the loop variable x declared at line 5, limited to that for loop. Trying to access that variable at line 7, which is out of scope of the variable x, causes a compile time error. So compilation fails due to error at line 7. Hence option D is correct. Options A and B are incorrect, since code fails to compile.

Reference:

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html>

### NEW QUESTION 125

Given the code fragment:

```
System.out.println(28 + 5 <= 4+ 29);  
System.out.println((28 + 5) <= (4 + 29));
```

What is the result?

- A. 28false29  
true
- B. 285 < 429  
true
- C. true  
true
- D. compilation fails

**Answer: C**

### NEW QUESTION 126

Given:

```
public class Equal {  
public static void main(String[] args) {  
String str1 = "Java";  
String[] str2 = {"J","a","v","a"};  
String str3 = "";  
for (String str : str2) {  
str3 = str3+str;  
}
```

```
boolean b1 = (str1 == str3);
boolean b2 = (str1.equals(str3));
System.out.print(b1+", "+b2);
```

What is the result?

- A. true, false
- B. false, true
- C. true, true
- D. false, false

**Answer:** B

**Explanation:**

`==` strict equality.  
`equals` compare state, not identity.

### NEW QUESTION 127

Given:

```
public class Test {
    static void dispResult(int[] num) {
        try {
            System.out.println(num[1] / (num[1] - num[2]));
        } catch(ArithmeticException e) {
            System.err.println("first exception");
        }
        System.out.println("Done");
    }
    public static void main(String[] args) {
        try {
            int [] arr = {100, 100};
            dispResult(arr);
        } catch(InvalidArgumentException e) {
            System.err.println("second exception");
        } catch(Exception e) {
            System.err.println("third exception");
        }
    }
}
```

What is the result?

- A. 0  
Done
- B. First Exception  
Done
- C. Second Exception
- D. Done  
Third Exception
- E. Third Exception

**Answer:** B

### NEW QUESTION 128

Given:

```
public class Marklist {
```

```
int num;
public static void graceMarks(Marklist obj4) {
obj4.num += 10;
}
public static void main(String[] args) {
MarkList obj1 = new MarkList();
MarkList obj2 = obj1;
MarkList obj1 = null;
obj2.num = 60;
graceMarks(obj2);
}
}
```

How many objects are created in the memory runtime?

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

**Answer:** B

**Explanation:**

obj1 and obj3.

when you do e2 = e1 you're copying object references - you're not making a copy of the object - and so the variables e1 and e2 will both point to the same object.

### NEW QUESTION 129

Given:

```
public class X implements Z {
    public String toString() {
        return "X";
    }
    Public static void main(String[] args) {
        Y myY = new Y();
        X myX = myY;
        Z myZ = myX;
        System.out.print(myX);
        System.out.print((Y)myX);
        System.out.print(myZ);
    }
}
class Y extends X {
    public String toString() {
        return "Y";
    }
}
```

What is the result?

- A. X XX
- B. X Y X
- C. Y Y X
- D. Y YY

**Answer:** D

**NEW QUESTION 130**

Given:

```
class Patient {  
    String name;  
    public Patient(String name) {  
        this.name = name;  
    }  
}
```

And the code fragment:

```
8. public class Test {  
9.     public static void main(String[] args) {  
10.         List ps = new ArrayList();  
11.         Patient p2 = new Patient("Mike");  
12.         ps.add(p2);  
13.  
14.         //insert code here  
15.  
16.         if(f >= 0) {  
17.             System.out.print("Mike Found");  
18.         }  
19.     }  
20. }
```

Which code fragment, when inserted at line 14, enables the code to print Mike Found?

- A. int f = ps.indexOf {new patient ("Mike")};
- B. int f = ps.indexOf (patient("Mike"));
- C. patient p = new Patient ("Mike");  
int f = pas.indexOf(P);
- D. int f = ps.indexOf(p2);

**Answer: C**

**NEW QUESTION 131**

Given:

```
public class Test {  
public static void main(String[] args) {  
try {  
String[] arr =new String[4];  
arr[1] = "Unix";  
arr[2] = "Linux";  
arr[3] = "Solarios";  
for (String var : arr) {  
System.out.print(var + " ");  
}  
} catch(Exception e) {  
System.out.print (e.getClass());  
}  
}
```

What is the result?

- A. Unix Linux Solaris

- B. Null Unix Linux Solaris
- C. Class java.lang.Exception
- D. Class java.lang.NullPointerException

**Answer:** B

**Explanation:**

null Unix Linux Solarios

The first element, arr[0], has not been defined.

### **NEW QUESTION 132**

Given:

```
public class Series {  
    private boolean flag;  
    public void displaySeries() {  
        int num = 2;  
        while(flag) {  
            if(num%7 == 0)  
                flag = false;  
            System.out.print(num);  
            Num += 2;  
        }  
    }  
    public static void main(String[] args) {  
        new Series().displaySeries();  
    }  
}
```

What is the result?

- A. 2 4 6 8 10 12
- B. 2 4 6 8 10 12 14
- C. Compilation fails
- D. The program prints multiple of 2 infinite times
- E. The program prints nothing

**Answer:** B

### **NEW QUESTION 133**

Which of the following can fill in the blank in this code to make it compile?

```
interface CanFly {  
    String type = "A";  
    Void fly();  
    _____ String getType() {  
        Return type;  
    }  
}
```

- A. abstract
- B. public
- C. default
- D. It will not compile with any as interfaces cannot have non abstract methods
- E. It will compile without filling the blank

**Answer:** C

**Explanation:**

From Java SE 8, we can use static and/or default methods in interfaces, but they should be non abstract methods. SO in this case using default in blank is completely legal. Hence option C is correct. Option A is incorrect as given method is not abstract, so can't use abstract there. Options B and E are incorrect as we can't have non abstract method interface if they are not default or static.

<https://docs.oracle.com/javase/tutorial/java/lambda/defaultmethods.html>

**NEW QUESTION 134**

Consider following method:

```
default void print() {  
}
```

Which statement is true?

- A. This method is invalid.
- B. This method can be used only in an interface.
- C. This method can return anything.
- D. This method can be used only in an interface or an abstract class.
- E. None of above.

**Answer: B****Explanation:**

Given method is declared as default method so we can use it only inside an interface. Hence option B is correct and option D is incorrect. Option A is incorrect as it is valid method. Option C is incorrect as return type is void, which means we can't return anything.

**NEW QUESTION 135**

Given:

```
public class MyFor3 {  
    public static void main(String[] args) {  
        int[] xx = null;  
        for(int ii : xx) {  
            System.out.println(ii);  
        }  
    }  
}
```

What is the result?

- A. Null
- B. Compilation fails
- C. An exception is thrown at runtime
- D. 0

**Answer: C****NEW QUESTION 136**

Given:

```
1. public class TestLoop {  
2.     public static void main(String[] args) {  
3.         float myarray[] = {10.20f, 20.30f, 30.40f, 50.60f};  
4.         int index = 0;  
5.         boolean isFound = false;  
6.         float key = 30.40f;  
7.         //insert code here  
8.         System.out.println(isFound);
```

```
9.    }
10. }
```

Which code fragment, when inserted at line 7, enables the code print true?

**Option A.**

```
while(key == myarray[index++]) {
    isFound == ture;
}
```

**Option B.**

```
while(index <= 4) {
    if(key == myarray[index]) {
        index++;
        isFound = true;
        break;
    }
}
```

**Option C.**

```
while(index++ < 5) {
    if(key == myarray[index]) {
        isFound = true;
    }
}
```

**Option D.**

```
while(index < 5) {
    if(key == myarray[index]) {
        isFound = true;
        break;
    }
    index++;
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer: A**

### NEW QUESTION 137

Given:

```
class Base {
public static void main(String[] args) {
System.out.println("Base " + args[2]);
}
}
public class Sub extends Base{
public static void main(String[] args) {
System.out.println("Overriden " + args[1]);
}
}
```

And the commands:

```
javac Sub.java
java Sub 10 20 30
```

What is the result?

- A. Base 30
- B. Overridden 20
- C. Overridden 20  
Base 30
- D. Base 30  
Overridden 20

**Answer:** B

### NEW QUESTION 138

Given:

```
class SpecialException extends Exception {
    public SpecialException(String message) {
        super(message);
        System.out.println(message);
    }
}
public class ExceptionTest {
    public static void main(String[] args) {
        try {
            doSomething();
        } catch(SpecialException e) {
            System.out.println(e);
        }
    }
    static void doSomething() throws SpecialException {
        int[] ages = new int[4];
        ages[4] = 17;
        doSomethingElse();
    }
    static void doSomethingElse() throws SpecialException {
        throw new SpecialException("Thrown at end of doSomething() method");
    }
}
```

What will be the output?

**Option A.**

SpecialException: Thrown at end of doSomething() method

**Option B.**

Error in thread "main" java.lang.ArrayIndexOutOfBoundsException

**Option C.**

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException:4  
at ExceptionTest.doSomething(ExceptionTest.java:13)

at ExceptionTest.main(ExceptionTest.java:4)

**Option D.**

SpecialException:Thrown at end of doSomething() method  
at ExceptionTest.doSomethingElse(ExceptionTest.java:16)  
at ExceptionTest.doSomething(ExceptionTest.java:13)  
at ExceptionTest.main(ExceptionTest.java:4)

- A. Option A
- B. Option B
- C. Option C

D. Option D

**Answer:** D

**NEW QUESTION 139**

Given the code fragments:

```
interface Contract { }
class Super implements Contract { }
class Sub extends Super { }
public class Ref {
    public static void main(String[] args) {
        List objs = new ArrayList();
        Contract c1 = new Super();
        Contract c2 = new Sub();      //line n1
        Super s1 = new Sub();
        objs.add(c1);
        objs.add(c2);
        objs.add(s1);      //line n2
        for(Object itm:objs) {
            System.out.println(itm.getClass().getName());
        }
    }
}
```

What is the result?

- A. Super  
Sub  
Sub
- B. Contract  
Contract  
Super
- C. Compilation fails at line n1
- D. Compilation fails at line n2

**Answer:** D

**NEW QUESTION 140**

Given:

```
public class Test {
    public static void main(String[] args) {
        Test ts = new Test();
        System.out.print(isAvailable + " ");
        isAvailable = ts.doStuff();
        System.out.println(isAvailable);
    }
    public static boolean doStuff() {
        return !isAvailable;
    }
    static boolean isAvailable = false;
}
```

What is the result?

- A. true true
- B. true false

- C. false true
- D. false false
- E. Compilation fails

**Answer:** E

#### **NEW QUESTION 141**

Given:

```
public class Msg {  
    public static String doMsg(char x) {  
        return "Good Day!";  
    }  
    public static String doMsg(int y) {  
        return "Good Luck!";  
    }  
    public static void main(String[] args) {  
        char x = 8;  
        int z = '8';  
        System.out.println(doMsg(x));  
        System.out.print(doMsg(z));  
    }  
}
```

What is the result?

- A. Good Day!  
Good Luck!
- B. Good Day!  
Good Day!
- C. Good Luck!  
Good Day!
- D. Good Luck!  
Good Luck!
- E. Compilation fails

**Answer:** E

#### **NEW QUESTION 142**

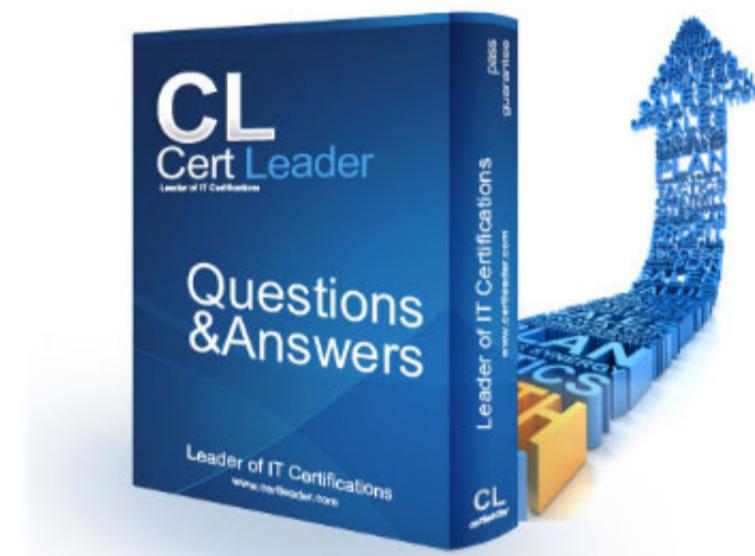
.....

**Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps**

## 1z0-808 Dumps

### Java SE 8 Programmer I

<https://www.certleader.com/1z0-808-dumps.html>



**NEW QUESTION 1**

Which one of the following code examples uses valid Java syntax?

- A.
- ```
public class Boat {  
  
    public static void main (String [] args) {  
        System.out.println ("I float.");  
    }  
}
```
- B.
- ```
public class Cake {  
    public static void main (String [] ) {  
        System.out.println ("Chocolate");  
    }  
}
```
- C.
- ```
public class Dog {  
    public void main (String [] args) {  
        System.out.println ("Squirrel.");  
    }  
}
```
- D.
- ```
public class Bank {  
    public static void main (String () args) {  
        System.out.println ("Earn interest.");  
    }  
}
```

- A. Option A  
B. Option B  
C. Option C  
D. Option D

**Answer:** A

**NEW QUESTION 2**

You are asked to create a method that accepts an array of integers and returns the highest value from that array.

Given the code fragment:

```
class Test{  
    public static void main(String[] args) {  
        int numbers[] = {12, 13, 42, 32, 15, 156, 23, 51, 12};  
        int[] keys = findMax(numbers);  
    }  
  
    /* line n1 */ {  
        int[] keys = new int[3];  
        /* code goes here*/  
        return keys;  
    }  
}
```

Which method signature do you use at line n1?

- A. public int findMax (int[] numbers)  
B. static int[] findMax (int[] max)  
C. static int findMax (int[] numbers)  
D. final int findMax (int[] )

**Answer: C****NEW QUESTION 3**

Given this code for a Planet object:

```
public class Planet {  
    public String name;  
    public int moons;  
  
    public Planet(String name, int moons) {  
        this.name = name;  
        this.moons = moons;  
    }  
}
```

And this method:

```
public static void main(String[] args){  
    Planet[] planets = {  
        new Planet("Mercury", 0),  
        new Planet("Venus", 0),  
        new Planet("Earth", 1),  
        new Planet("Mars", 2)  
    };  
  
    System.out.println(planets);  
    System.out.println(planets[2].name);  
    System.out.println(planets[2].moons);  
}
```

What is the output?

A  
planets  
Earth  
1

B  
[LPlanets.Planet;@15db9742  
Earth  
1

C  
[LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
1

D  
[LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
[LPlanets.Moon;@7852e922

E  
[LPlanets.Planet;@15db9742  
Venus  
0

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer: C****NEW QUESTION 4**

Given the code fragment:

```
public static void main(String[] args) {  
    Short s1 = 200;  
    Integer s2 = 400;  
    Long s3 = (long) s1 + s2;           //line n1  
    String s4 = (String) (s3 * s2);     //line n2  
    System.out.println("Sum is " + s4);  
}
```

What is the result?

- A. Sum is 600
- B. Compilation fails at line n1.
- C. Compilation fails at line n2.
- D. A ClassCastException is thrown at line n1.
- E. A ClassCastException is thrown at line n2.

**Answer:** C

#### NEW QUESTION 5

Given the code fragment:

```
public static void main(String[] args) {  
    int data[] = {2010, 2013, 2014, 2015, 2014};  
    int key = 2014;  
    int count = 0;  
    for (int e: data) {  
        if (e != key) {  
            continue;  
            count++;  
        }  
    }  
    System.out.print(count + " Found");  
}
```

What is the result?

- A. Compilation fails.
- B. 0 Found
- C. 1 Found
- D. 3 Found

**Answer:** A

#### NEW QUESTION 6

Given the code from the Greeting.Java file:

```
public class Greeting {  
    public static void main(String[] args) {  
        System.out.println("Hello " + args[0]);  
    }  
}
```

Which set of commands prints Hello Duke in the console?

- A) javac Greeting  
java Greeting Duke
- B) javac Greeting.java Duke  
java Greeting
- C) javac Greeting.java  
java Greeting Duke
- D) javac Greeting.java  
java Greeting.class Duke

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** C

#### NEW QUESTION 7

Given:

```
public class Fieldinit {  
    char c;  
    boolean b;  
    float f;  
    void printAll() {  
        System.out.println ("c = " + c);  
        System.out.println ("b = " + b);  
        System.out.println ("f = " + f);  
    }  
    public static void main (String [] args) {  
        FieldInit f = new FieldInit ();  
        f.printAll ();  
    }  
}
```

What is the result?

A

```
c=  
b = false  
f = 0.0
```

B

```
c= null  
b = true  
f = 0.0
```

C

```
c=0  
b = false  
f = 0.0f
```

D

```
c= null  
b = false  
f = 0.0F
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

#### NEW QUESTION 8

Given:

```
public class MyClass {  
    public static void main(String[] args) {  
        String s = "Java SE 8 1";  
        int len = s.trim().length();  
        System.out.print(len);  
    }  
}
```

What is the result?

- A. Compilation fails.
- B. 11
- C. 8
- D. 9
- E. 10

**Answer:** B

**NEW QUESTION 9**

Given the code fragment:

```
public class Employee {  
    String name;  
    boolean contract;  
    double salary;  
    Employee() {  
        // line n1  
    }  
    public String toString(){  
        return name + ":" + contract + ":" + salary;  
    }  
    public static void main(String[] args) {  
        Employee e = new Employee();  
        // line n2  
        System.out.print(e);  
    }  
}
```

Which two modifications, when made independently, enable the code to print Joe:true: 100.0? (Choose two.)

A) Replace line n2 with:

```
e.name = "Joe";  
e.contract = true;  
e.salary = 100;
```

B) Replace line n2 with:

```
this.name = "Joe";  
this.contract = true;  
this.salary = 100;
```

C) Replace line n1 with:

```
this.name = new String("Joe");  
this.contract = new Boolean(true);  
this.salary = new Double(100);
```

D) Replace line n1 with:

```
name = "Joe";  
contract = TRUE;  
salary = 100.0f;
```

E) Replace line n1 with:

```
this("Joe", true, 100);
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

**Answer:** AC

**NEW QUESTION 10**

Given the code fragment:

```
LocalDateTime dt = LocalDateTime.of(2014, 7, 31, 1, 1);  
dt.plusDays(30);  
dt.plusMonths(1);  
System.out.println(dt.format(DateTimeFormatter.ISO_DATE_TIME));
```

What is the result?

A. An exception is thrown at runtime

B. 2014-07-31T01:01:00

C. 2014-07-31

D. 2014-09-30T00:00:00

**Answer:** B

**NEW QUESTION 11**

Which statement is true about Java byte code?

A. It can run on any platform.

B. It can run on any platform only if it was compiled for that platform.

C. It can run on any platform that has the Java Runtime Environment.

D. It can run on any platform that has a Java compiler.

E. It can run on any platform only if that platform has both the Java Runtime Environment and a Java compiler.

**Answer:** D**Explanation:**

Java bytecodes help make "write once, run anywhere" possible. You can compile your program into bytecodes on any platform that has a Java compiler. The bytecodes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

**NEW QUESTION 12**

Given:

Base.java:

```
class Base {  
    public void test(){  
        System.out.println("Base ");  
    }  
}
```

DerivedA.java:

```
class DerivedA extends Base {  
    public void test(){  
        System.out.println("DerivedA ");  
    }  
}
```

DerivedB.java:

```
class DerivedB extends DerivedA {  
    public void test(){  
        System.out.println("DerivedB ");  
    }  
    public static void main(String[] args) {  
        Base b1 = new DerivedB();  
        Base b2 = new DerivedA();  
        Base b3 = new DerivedB();  
        Base b4 = b3;  
        b1 = (Base) b2;  
        b1.test();  
        b4.test();  
    }  
}
```

What is the result?

- A. BaseDerivedA
- B. BaseDerivedB
- C. DerivedBDerivedB
- D. DerivedBDerivedA
- E. A ClassCastException is thrown at runtime.

**Answer:** D**NEW QUESTION 13**

Given the code fragment:

```
public static void main(String[] args) {  
    LocalDate date = LocalDate.of(2012, 1, 30);  
    date.plusDays(10);  
    System.out.println(date);  
}
```

What is the result?

- A. 2012-02-10
- B. 2012-01-30
- C. 2012-02-10 00:00
- D. A DateTimeException is thrown at runtime.

**Answer:** C**NEW QUESTION 14**

Given the code fragment:

```
public static void main(String[] args) {
    StringBuilder sb = new StringBuilder("Java");
    String s = "Java";

    if (sb.toString().equals(s.toString())) {
        System.out.println("Match 1");
    } else if (sb.equals(s)) {
        System.out.println("Match 2");
    } else {
        System.out.println("No Match");
    }
}
```

What is the result?

- A. Match 1
- B. Match 2
- C. No Match
- D. A NullPointerException is thrown at runtime.

**Answer:** A

#### NEW QUESTION 15

Given:

```
class Caller {
    private void init () {
        System.out.println("Initialized");
    }

    private void start () {
    init();
    System.out.println("Started");
    }
}

public class TestCall {
    public static void main(String[] args) {
        Caller c - new Caller();
        c.start();
        c.init();
    }
}
```

What is the result?

- A. An exception is thrown at runtime.
- B. InitializedStartedInitialized
- C. InitializedStarted
- D. Compilation fails.

**Answer:** D

#### NEW QUESTION 16

Which three statements describe the object-oriented features of the Java language? (Choose three.)

- A. Objects cannot be reused.
- B. A subclass must override the methods from a superclass.
- C. Objects can share behaviors with other objects.
- D. A package must contain a main class.
- E. Object is the root class of all other objects.
- F. A main method must be declared in every class.

**Answer:** BCF

#### NEW QUESTION 17

Which statement will empty the contents of a StringBuilder variable named sb?

- A. s
- B. deleteAll();
- C. s
- D. delete (0, s
- E. size ();
- F. s
- G. delete (0, s

H. length () ;  
I. s  
J. removeAll () ;

**Answer: C**

**NEW QUESTION 18**

Given the code fragment:

```
public static void main(String[] args) {  
    int[][] arr = new int [2] [4];  
    arr[0] = new int []{1, 3, 5, 7};  
    arr[1] = new int []{1, 3};  
    for (int[] a : arr) {  
        for (int i : a) {  
            System.out.print(i+ " ");  
        }  
        System.out.println();  
    }  
}
```

What is the result?

- A Compilation fails.
- B  
1 3  
1 3
- C  
1 3  
followed by an ArrayIndexOutOfBoundsException
- D  
1 3  
1 3 0 0
- E  
1 3 5 7  
1 3

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer: E**

**Explanation:**

Your Code ...

```
1- public class MyClass {
2-     public static void main (String [] args) {
3-         int [][] arr =new int [2] [4];
4-         arr[0] = new int [] {1, 3, 5, 7};
5-         arr[1] = new int [] {1, 3};
6-         for (int [] a : arr) {
7-             for (int i : a) {
8-                 System.out.print(i+ " ");
9-             }
10-            System.out.println ();
11-        }
12-    }
13- }
```

External Libraries ...

CommandLine Arguments ...

Interactive mode :  OFF      Version:

Stdin Inputs...

Result...  
CPU Time: 0.13 sec(s), Memory: 30680 kilobyte(s)      compiled and executed in 0.705 sec(s)

```
1 3 5 7
1 3
```

**NEW QUESTION 19**

## Thank You for Trying Our Product

\* **100% Pass or Money Back**

All our products come with a 90-day Money Back Guarantee.

\* **One year free update**

You can enjoy free update one year. 24x7 online support.

\* **Trusted by Millions**

We currently serve more than 30,000,000 customers.

\* **Shop Securely**

All transactions are protected by VeriSign!

**100% Pass Your 1z0-808 Exam with Our Prep Materials Via below:**

<https://www.certleader.com/1z0-808-dumps.html>



**1z0-808**

Number: 1z0-808  
Passing Score: 800  
Time Limit: 120 min  
File Version: 14.1

**Exam code: 1z0-808**

**Exam name: Java SE 8 Programmer I**

**Version 14.1**

**Exam A****QUESTION 1**

What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?

- A. Encapsulation
- B. Inheritance
- C. Abstraction
- D. Instantiation
- E. Polymorphism

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Explanation:

Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

Reference: [http://www.tutorialspoint.com/java/java\\_access\\_modifiers.htm](http://www.tutorialspoint.com/java/java_access_modifiers.htm)

**QUESTION 2**

Given the code fragment:

```
abstract class Planet {  
    protected void revolve() { //line n1  
    }  
  
    abstract void rotate(); //line n2  
}  
  
class Earth extends Planet {  
    void revolve() { //line n3  
    }  
  
    protected void rotate() { //line n4  
    }  
}
```

Which two modifications, made independently, enable the code to compile?

- A. Make the method at line n1 public.
- B. Make the method at line n2 public.
- C. Make the method at line n3 public.
- D. Make the method at line n3 protected.
- E. Make the method at line n4 public.

**Correct Answer:** BC

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 3**

Given:

```
class Vehicle {  
    String type = "4W";  
    int maxSpeed = 100;  
  
    Vehicle(String type, int maxSpeed) {  
        this.type = type;  
        this.maxSpeed = maxSpeed;  
    }  
}  
  
class Car extends Vehicle {  
    String trans;  
  
    Car(String trans) {           //line n1  
        this.trans = trans;  
    }  
  
    Car(String type, int maxSpeed, String trans) {  
        super(type, maxSpeed);  
        this(trans);           //line n2  
    }  
}
```

And given the code fragment:

```
7. Car c1 = new Car("Auto");  
8. Car c2 = new Car("4W", 150, "Manual");  
9. System.out.println(c1.type + " " + c1.maxSpeed + " " + c1.trans);  
10. System.out.println(c2.type + " " + c2.maxSpeed + " " + c2.trans);
```

What is the result?

- A. 4W 100 Auto  
 4W 150 Manual
- B. Null 0 Auto  
 4W 150 Manual

- C. Compilation fails only at line n1
- D. Compilation fails only at line n2
- E. Compilation fails at both line n1 and line n2

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 4**

Given the code fragment:

```
1. class X {  
2.     public void printFileContent() {  
3.         /* code goes here */  
4.         throw new IOException();  
5.     }  
6. }  
7. public class Test {  
8.     public static void main(String[] args) {  
9.         X xobj = new X();  
10.        xobj.printFileContent();  
11.    }  
12. }
```

Which two modifications should you make so that the code compiles successfully?

- A) Replace line 8 with `public static void main(String[] args) throws Exception {`
  - B) Replace line 10 with:  

```
try {
    xobj.printFileContent();
}
catch(Exception e) { }
catch(IOException e) { }
```
  - C) Replace line 2 with `public void printFileContent() throws IOException {`
  - D) Replace line 4 with `throw IOException("Exception raised");`
  - E) At line 11, insert `throw new IOException();`
- A. Option A  
B. Option B  
C. Option C  
D. Option D  
E. Option E

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 5**

Given the following two classes:

```
public class Customer {
    ElectricAccount acct = new ElectricAccount();

    public void useElectricity(double kWh) {
        acct.addKWh(kWh);
    }
}

public class ElectricAccount {
    private double kWh;
    private double rate = 0.07;
    private double bill;

    //line n1
}
```

How should you write methods in the ElectricAccount class at line n1 so that the member variable bill is always equal to the value of the member variable kWh multiplied by the member variable rate?

Any amount of electricity used by a customer (represented by an instance of the customer class) must contribute to the customer's bill (represented by the member variable bill) through the method useElectricity method. An instance of the customer class should never be able to tamper with or decrease the value of the member variable bill.

- A) 

```
public void addKWh(double kWh) {
    this.kWh += kWh;
    this.bill = this.kWh*this.rate;
}
```
- B) 

```
public void addKWh(double kWh) {
    if (kWh > 0){
        this.kWh += kWh;
        this.bill = this.kWh * this.rate;
    }
}
```
- C) 

```
private void addKWh(double kWh) {
    if (kWh > 0) {
        this.kWh += kWh;
        this.bill = this.kWh*this.rate;
    }
}
```
- D) 

```
public void addKWh(double kWh) {
    if(kWh > 0) {
        this.kWh += kWh;
        setBill(this.kWh);
    }
}
public void setBill(double kWh) {
    bill = kWh*rate;
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** A

Section: (none)

Explanation

Explanation/Reference:

**QUESTION 6**

Given the code fragment:

```
public static void main(String[] args) {
    StringBuilder sb = new StringBuilder(5);
    String s = "";

    if (sb.equals(s)) {
        System.out.println("Match 1");
    } else if (sb.toString().equals(s.toString())) {
        System.out.println("Match 2");
    } else {
        System.out.println("No Match");
    }
}
```

What is the result?

- A. Match 1
- B. Match 2
- C. No Match
- D. A NullPointerException is thrown at runtime.

Correct Answer: B

Section: (none)

Explanation

Explanation/Reference:

**QUESTION 7**

Given:

```
interface Readable {
    public void readBook();
    public void setBookMark();
}

abstract class Book implements Readable { // line n1
    public void readBook() { }
    // line n2
}

class EBook extends Book { // line n3
    public void readBook() { }
    // line n4
}
```

Which option enables the code to compile?

- A) Replace the code fragment at line n1 with:

```
    class Book implements Readable {
```

- B) At line n2 insert:

```
    public abstract void setBookMark();
```

- C) Replace the code fragment at line n3 with:

```
    abstract class EBook extends Book {
```

- D) At line n4 insert:

```
    public void setBookMark() { }
```

A. Option A

B. Option B

C. Option C

D. Option D

**Correct Answer: C**

**Section: (none)**

**Explanation****Explanation/Reference:****QUESTION 8**

Given:

```
public static void main(String[] args) {  
    String ta = "A ";  
    ta = ta.concat("B ");  
    String tb = "C ";  
    ta = ta.concat(tb);  
    ta.replace('C', 'D');  
    ta = ta.concat(tb);  
    System.out.println(ta);  
}
```

What is the result?

- A. A B C D
- B. A C D
- C. A B C C
- D. A B D
- E. A B D C

**Correct Answer:** D**Section:** (none)**Explanation****Explanation/Reference:****QUESTION 9**

Given:

```
class CD {  
    int r;  
    CD(int r){  
        this.r=r;  
    }  
}  
  
class DVD extends CD {  
    int c;  
    DVD(int r, int c) {  
        // line n1  
    }  
}
```

And given the code fragment:

```
DVD dvd = new DVD(10,20);
```

Which code fragment should you use at line n1 to instantiate the dvd object successfully?

- A) super.r = r;  
 this.c = c;
- B) super(r);  
 this(c);
- C) super(r);  
 this.c = c;
- D) this.c = r;  
 super(c);

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** C  
**Section:** (none)  
**Explanation**

**Explanation/Reference:**

**QUESTION 10**

Given the code fragment:

```
int a[] = {1, 2, 3, 4, 5};  
for(XXX) {  
    System.out.print(a[e]);  
}
```

Which option can replace xxx to enable the code to print 135?

- A. int e = 0; e <= 4; e++
- B. int e = 0; e < 5; e += 2
- C. int e = 1; e <= 5; e += 1
- D. int e = 1; e < 5; e+=2

**Correct Answer:** D  
**Section:** (none)  
**Explanation**

**Explanation/Reference:**

**QUESTION 11**

Which statement best describes encapsulation?

- A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects.
- B. Encapsulation ensures that classes can be designed so that their methods are inheritable.
- C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.
- D. Encapsulation ensures that classes can be designed so that if a method has an argument MyType x, any subclass of MyType can be passed to that method.

**Correct Answer:** D

Section: (none)

Explanation

Explanation/Reference:

**QUESTION 12**

Given the code fragment from three files:

SalesMan.java:

```
package sales;
public class SalesMan { }
```

Product.java:

```
package sales.products;
public class Product { }
```

Market.java:

```
1. package market;
2. // insert code here
3. public class USMarket {
4.     SalesMan sm;
5.     Product p;
6. }
```

Which code fragment, when inserted at line 2, enables the code to compile?

- A) import sales.\*;
- B) import java.sales.products.\*;
- C) import sales;  
    import sales.products;
- D) import sales.\*;  
    import products.\*;
- E) import sales.\*;  
    import sales.products.\*;

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 13**

Given the following class:

```
public class CheckingAccount {  
    public int amount;  
    public CheckingAccount(int amount) {  
        this.amount = amount;  
    }  
    public int getAmount () {  
        return amount;  
    }  
    public void changeAmount (int x) {  
        amount += x;  
    }  
}
```

And given the following main method, located in another class:

```
public static void main(String[] args) {  
    CheckingAccount acct = new CheckingAccount((int)(Math.random()*1000));  
    //line n1  
    System.out.println(acct.getAmount());  
}
```

Which three lines, when inserted independently at line n1, cause the program to print a 0 balance?

- A. this.amount = 0;
- B. amount = 0;
- C. acct(0);
- D. acct.amount = 0;
- E. acct.getAmount() = 0;
- F. acct.changeAmount(0);
- G. acct.changeAmount(-acct.amount);
- H. acct.changeAmount(-acct.getAmount());

**Correct Answer:** ACD

**Section:** (none)

**Explanation**

**Explanation/Reference:****QUESTION 14**

Given the code fragment:

```
String shirts[][] = new String[2][2];
shirts[0][0] = "red";
shirts[0][1] = "blue";
shirts[1][0] = "small";
shirts[1][1] = "medium";
```

Which code fragment prints red: blue: small: medium?

C A) for (int index = 1; index < 2; index++) {  
 for (int idx = 1; idx < 2; idx++) {  
 System.out.print(shirts[index][idx] + ":" );  
 }  
}  
  
C B) for (int index = 0; index < 2; ++index) {  
 for (int idx = 0; idx < index; ++idx) {  
 System.out.print(shirts[index][idx] + ":" );  
 }  
}  
  
C C) for (String c : colors) {  
 for (String s : sizes) {  
 System.out.println(s + ":" );  
 }  
}  
  
C D) for (int index = 0; index < 2;) {  
 for (int idx = 0; idx < 2;) {  
 System.out.print(shirts[index][idx] + ":" );  
 idx++;  
 }  
 index++;  
}

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:****QUESTION 15**

Given the code fragment:

```
int x = 100;
int a = x++;
int b = ++x;
int c = x++;
int d = (a < b) ? (a < c) ? a: (b < c )? b: c;
System.out.println(d);
```

What is the result?

- A. 100
- B. 101
- C. 102
- D. 103
- E. Compilation fails

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:****QUESTION 16**

Given:

```
public class Test {  
  
    public static void main(String[] args) {  
  
        String[][] chs = new String[2][];  
        chs[0] = new String[2];  
        chs[1] = new String[5];  
        int i = 97;  
  
        for (int a = 0; a < chs.length; a++) {  
            for (int b = 0; b < chs.length; b++) {  
                chs[a][b] = "" + i;  
                i++;  
            }  
        }  
  
        for (String[] ca : chs) {  
            for (String c : ca) {  
                System.out.print(c + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

What is the result?

- A. 91 98  
99 100 null null null
- B. 91 98  
99 100 101 102 103
- C. Compilation rails.
- D. A NullPointerException is thrown at runtime.
- E. An ArrayIndexOutOfBoundsException is thrown at runtime.

**Correct Answer:** D

**Section:** (none)

## Explanation

### Explanation/Reference:

#### QUESTION 17

Given the code fragment:

```
public class Employee {  
    String name;  
    boolean contract;  
    double salary;  
    Employee() {  
        // line n1  
    }  
    public String toString(){  
        return name + ":" + contract + ":" + salary;  
    }  
    public static void main(String[] args) {  
        Employee e = new Employee();  
        // line n2  
        System.out.print(e);  
    }  
}
```

Which two modifications, when made independently, enable the code to print joe:true: 100.0?

A) Replace line n2 with:

```
e.name = "Joe";  
e.contract = true;  
e.salary = 100;
```

B) Replace line n2 with:

```
this.name = "Joe";  
this.contract = true;  
this.salary = 100;
```

C) Replace line n1 with:

```
this.name = new String("Joe");  
this.contract = new Boolean(true);  
this.salary = new Double(100);
```

D) Replace line n1 with:

```
name = "Joe";  
contract = TRUE;  
salary = 100.0f;
```

E) Replace line n1 with:

```
this("Joe", true, 100);
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 18

Given the code fragment:

```
public static void main(String[] args) {  
    List<String> names = new ArrayList<>();  
    names.add("Robb");  
    names.add("Bran");  
    names.add("Rick");  
    names.add("Bran");  
  
    if (names.remove("Bran")) {  
        names.remove("Jon");  
    }  
    System.out.println(names);  
}
```

What is the result?

- A. [Robb, Rick, Bran]
- B. [Robb, Rick]
- C. [Robb, Bran, Rick, Bran]
- D. An exception is thrown at runtime.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 19

Given:

```
class A {
    public A(){
        System.out.print("A ");
    }
}

class B extends A{
    public B() //line n1
        System.out.print("B ");
    }
}

class C extends B{

    public C() //line n2
        System.out.print("C ");
    }
    public static void main(String[] args) {
        C c = new C();
    }
}
```

What is the result?

- A. C B A
- B. C
- C. A B C
- D. Compilation fails at line n1 and line n2

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

## QUESTION 20

Given:

```
class X {  
    static int i;  
    int j;  
    public static void main(String[] args) {  
        X x1 = new X();  
        X x2 = new X();  
        x1.i = 3;  
        x1.j = 4;  
        x2.i = 5;  
        x2.j = 6;  
        System.out.println(  
            x1.i + " " +  
            x1.j + " " +  
            x2.i + " " +  
            x2.j);  
    }  
}
```

What is the result?

- A. 3 4 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 4 6

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 21

Given the code fragment:

```
1. public class Test {  
2.     public static void main(String[] args) {  
3.         /* insert code here */  
4.         array[0]=10;  
5.         array[1]=20;  
6.         System.out.print(array[0]+":"+array[1]);  
7.     }  
8. }
```

Which code fragment, when inserted at line 3, enables the code to print 10:20?

- A. int[] array n= new int[2];
- B. int[] array;  
array = int[2];
- C. int array = new int[2];
- D. int array [2];

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 22**

Given the code fragment:

```
public static void main(String[] args) {
    String[] arr = {"A", "B", "C", "D"};
    for (int i = 0; i < arr.length; i++) {
        System.out.print(arr[i] + " ");
        if (arr[i].equals("C")) {
            continue;
        }
        System.out.println("Work done");
        break;
    }
}
```

What is the result?

- A. A B C Work done
- B. A B C D Work done
- C. A Work done
- D. Compilation fails

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 23

Which three are advantages of the Java exception mechanism?

- A. Improves the program structure because the error handling code is separated from the normal program function
- B. Provides a set of standard exceptions that covers all the possible errors
- C. Improves the program structure because the programmer can choose where to handle exceptions
- D. Improves the program structure because exceptions must be handled in the method in which they occurred
- E. Allows the creation of new exceptions that are tailored to the particular program being created

**Correct Answer:** ACD

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Reference: <http://javajee.com/introduction-to-exceptions-in-java>

**QUESTION 24**

Given the code from the Greeting.Java file:

```
public class Greeting {  
    public static void main(String[] args) {  
        System.out.println("Hello " + args[0]);  
    }  
}
```

Which set of commands prints Hello Duke in the console?

- A) javac Greeting  
java Greeting Duke
- B) javac Greeting.java Duke  
java Greeting
- C) javac Greeting.java  
java Greeting Duke
- D) javac Greeting.java  
java Greeting.class Duke

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 25**

Given:

```
class Alpha {  
    int ns;  
    static int s;  
    Alpha(int ns) {  
        if (s < ns) {  
            s = ns;  
            this.ns = ns;  
        }  
    }  
    void doPrint() {  
        System.out.println("ns = " + ns + " s = " + s);  
    }  
}
```

And,

```
public class TestA {  
    public static void main(String[] args) {  
        Alpha ref1 = new Alpha(50);  
        Alpha ref2 = new Alpha(125);  
        Alpha ref3 = new Alpha(100);  
        ref1.doPrint();  
        ref2.doPrint();  
        ref3.doPrint();  
    }  
}
```

What is the result?

- A) ns = 50 s = 125  
ns = 125 s = 125  
ns = 100 s = 125
- B) ns = 50 s = 125  
ns = 125 s = 125  
ns = 0 s = 125
- C) ns = 50 s = 50  
ns = 125 s = 125  
ns = 100 s = 100
- D) ns = 50 s = 50  
ns = 125 s = 125  
ns = 0 s = 125

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 26**

Given the code fragment:

```
public static void main(String[] args) {  
    int ii = 0;  
    int jj = 7;  
    for (ii = 0; ii < jj - 1; ii = ii + 2) {  
        System.out.print(ii + " ");  
    }  
}
```

What is the result?

- A. 2 4
- B. 0 2 4 6
- C. 0 2 4
- D. Compilation fails

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 27

Given the code fragment:

```
LocalDate date1 = LocalDate.now();  
LocalDate date2 = LocalDate.of(2014, 6, 20);  
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);  
System.out.println("date1 = " + date1);  
System.out.println("date2 = " + date2);  
System.out.println("date3 = " + date3);
```

Assume that the system date is June 20, 2014. What is the result?

- A) date1 = 2014-06-20  
date2 = 2014-06-20  
date3 = 2014-06-20
- B) date1 = 06/20/2014  
date2 = 2014-06-20 [  
date3 = Jun 20, 2014
- C) Compilation fails.
- D) A DateParseException is thrown at runtime.

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 28

Given the code fragment:

```
7. StringBuilder sb1 = new StringBuilder("Duke");  
8. String str1 = sb1.toString();  
9. // insert code here  
10. System.out.print(str1 == str2);
```

Which code fragment, when inserted at line 9, enables the code to print true?

- A. String str2 = str1;
- B. String str2 = new String (str1);
- C. String str2 = sb1. toString ();

D. String str2 = "Duke";

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 29**

Given the code fragment:

```
public class Test {  
  
    static int count = 0;  
    int i = 0;  
  
    public void changeCount() {  
        while (i < 5) {  
            i++;  
            count++;  
        }  
    }  
  
    public static void main(String[] args) {  
        Test check1 = new Test();  
        Test check2 = new Test();  
        check1.changeCount();  
        check2.changeCount();  
        System.out.print(check1.count + " : " + check2.count);  
    }  
}
```

What is the result?

A. 10 : 10

B. 5 : 5

C. 5 : 10

D. Compilation fails

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 30

Given the code fragment:

```
public static void main(String[] args) {  
    double discount = 0;  
    int qty = Integer.parseInt(args[0]);  
    //line n1;  
}
```

And given the requirements:

If the value of the qty variable is greater than or equal to 90, discount = 0.5 If the value of the qty variable is between 80 and 90, discount = 0.2 Which two code fragments can be independently placed at line n1 to meet the requirements?

- A) if (qty >= 90) { discount = 0.5; }  
    if (qty > 80 && qty < 90) { discount = 0.2; }
- B) discount = (qty >= 90) ? 0.5 : 0;  
    discount = (qty > 80) ? 0.2 : 0;
- C) discount = (qty >= 90) ? 0.5 : (qty > 80)? 0.2 : 0;
- D) if (qty > 80 && qty < 90) {  
        discount = 0.2;  
    } else {  
        discount = 0;  
    }  
    if (qty >= 90) {  
        discount = 0.5;  
    } else {  
        discount = 0;  
    }
- E) discount = (qty > 80) ? 0.2 : (qty >= 90) ? 0.5 : 0;

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 31**

Given:

```
public class Test {  
  
    public static void main(String[] args) {  
        if (args[0].equals("Hello") ? false : true) {  
            System.out.println("Success");  
        } else {  
            System.out.println("Failure");  
        }  
    }  
}
```

And given the commands:

```
javac Test.java  
Java Test Hello
```

What is the result?

- A. Success
- B. Failure
- C. Compilation fails.
- D. An exception is thrown at runtime

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 32

Which three statements describe the object-oriented features of the Java language?

- A. Objects cannot be reused.
- B. A subclass can inherit from a superclass.
- C. Objects can share behaviors with other objects.
- D. A package must contain more than one class.
- E. Object is the root class of all other objects.

F. A main method must be declared in every class.

**Correct Answer:** BCF

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 33

Given:

```
package p1;
public class Acc {
    int p;
    private int q;
    protected int r;
    public int s;
}
```

Test.java:

```
package p2;
import p1.Acc;
public class Test extends Acc {
    public static void main(String[] args) {
        Acc obj = new Test();
    }
}
```

Which statement is true?

- A. Both p and s are accessible by obj.
- B. Only s is accessible by obj.
- C. Both r and s are accessible by obj.
- D. p, r, and s are accessible by obj.

**Correct Answer:** C

Section: (none)  
Explanation

Explanation/Reference:

**QUESTION 34**

Given:

Base.java:

```
class Base {  
    public void test(){  
        System.out.println("Base ");  
    }  
}
```

DerivedA.java:

```
class DerivedA extends Base {  
    public void test(){  
        System.out.println("DerivedA ");  
    }  
}
```

DerivedB.java:

```
class DerivedB extends DerivedA {  
    public void test(){  
        System.out.println("DerivedB ");  
    }  
    public static void main(String[] args) {  
        Base b1 = new DerivedB();  
        Base b2 = new DerivedA();  
        Base b3 = new DerivedB();  
        b1 = (Base) b3;  
        Base b4 = (DerivedA) b3;  
        b1.test();  
        b4.test();  
    }  
}
```

What is the result?

- A. Base

- DerivedA
- B. Base
- DerivedB
- C. DerivedB
- DerivedB
- D. DerivedB
- DerivedA
- E. A classcast Except ion is thrown at runtime.

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 35

Given the code fragment:

```
public static void main(String[] args) {  
    ArrayList myList = new ArrayList();  
    String[] myArray;  
    try {  
        while (true) {  
            myList.add("My String");  
        }  
    }  
    catch (RuntimeException re) {  
        System.out.println("Caught a RuntimeException");  
    }  
    catch (Exception e) {  
        System.out.println("Caught an Exception");  
    }  
    System.out.println("Ready to use");  
}
```

What is the result?

- A. Execution terminates in the first catch statement, and caught a RuntimeException is printed to the console.
- B. Execution terminates In the second catch statement, and caught an Exception is printed to the console.
- C. A runtime error is thrown in the thread "main".

- D. Execution completes normally, and Ready to use is printed to the console.
- E. The code fails to compile because a throws keyword is required.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 36**

Given:

```
System.out.println("5 + 2 = " + 3 + 4);  
System.out.println("5 + 2 = " + (3 + 4));
```

What is the result?

- A) 5 + 2 = 34  
5 + 2 = 34
- B) 5 + 2 + 3 + 4  
5 + 2 = 7
- C) 7 = 7  
7 + 7
- D) 5 + 2 = 34  
5 + 2 = 7

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** B

**Section:** (none)

**Explanation****Explanation/Reference:****QUESTION 37**

Given the code fragments:

Person.java:

```
public class Person {  
    String name;  
    int age;  
  
    public Person(String n, int a) {  
        name = n;  
        age = a;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getAge() {  
        return age;  
    }  
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {  
    for (Person p : list) {  
        if (predicate.test(p)) {  
            System.out.println(p.name + " ");  
        }  
    }  
}  
  
public static void main(String[] args) {  
    List<Person> iList = Arrays.asList(new Person("Hank", 45),  
                                         new Person("Charlie", 40),  
                                         new Person("Smith", 38));  
    //line n1  
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

- A. checkAge (iList, () -> p. get Age ( ) > 40);
- B. checkAge(iList, Person p -> p.getAge( ) > 40);
- C. checkAge (iList, p -> p.getAge ( ) > 40);
- D. checkAge(iList, (Person p) -> { p.getAge() > 40; });

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 38

Given the code fragment:

```
public static void main(String[] args) {
    String[][] arr = {{"A", "B", "C"}, {"D", "E"}};
    for (int i = 0; i < arr.length; i++) {
        for (int j = 0; j < arr[i].length; j++) {
            System.out.print(arr[i][j] + " ");
            if (arr[i][j].equals("B")) {
                break;
            }
        }
        continue;
    }
}
```

What is the result?

- A. A B C
- B. A B C D E
- C. A B D E
- D. Compilation fails.

**Correct Answer:** C  
**Section:** (none)  
**Explanation**

**Explanation/Reference:**

**QUESTION 39**

Given the code fragment:

```
public static void main(String[] args) {  
    String str = " ";  
    str.trim();  
    System.out.println(str.equals("") + " " + str.isEmpty());  
}
```

What is the result?

- A. true true
- B. true false
- C. false false
- D. false true

**Correct Answer:** C  
**Section:** (none)  
**Explanation**

**Explanation/Reference:**

**QUESTION 40**

Given the code fragment:

```
public class App {  
    public static void main(String[] args) {  
        String str1 = "Java";  
        String str2 = new String("java");  
        //line n1  
        {  
            System.out.println("Equal");  
        } else {  
            System.out.println("Not Equal");  
        }  
    }  
}
```

Which code fragment, when inserted at line n1, enables the App class to print Equal?

- A) String str3 = str2;  
if (str1 == str3)
- B) if (str1.equalsIgnoreCase(str2))
- C) String str3 = str2;  
if (str1.equals(str3))
- D) if (str1.toLowerCase() == str2.toLowerCase())

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** A  
**Section:** (none)  
**Explanation**

**Explanation/Reference:**

**QUESTION 41**

Given:

```
public class SumTest {  
  
    public static void doSum(Integer x, Integer y) {  
        System.out.println("Integer sum is " + (x + y));  
    }  
  
    public static void doSum(double x, double y) {  
        System.out.println("double sum is " + (x + y));  
    }  
  
    public static void doSum(float x, float y) {  
        System.out.println("float sum is " + (x + y));  
    }  
  
    public static void doSum(int x, int y) {  
        System.out.println("int sum is " + (x + y));  
    }  
  
    public static void main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

What is the result?

- A) int sum is 30  
float sum is 30.0
- B) int sum is 30  
double sum is 30
- C) Integer sum is 30  
double sum is 30.0
- D) Integer sum is 30  
float sum is 30.0

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 42

Given the code fragment:

```
String[] strs = new String[2];
int idx = 0;
for (String s : strs) {
    strs[idx].concat(" element " + idx);
    idx++;
}
for (idx = 0; idx < strs.length; idx++) {
    System.out.println(strs[idx]);
}
```

What is the result?

- A. Element 0  
Element 1
- B. Null element 0  
Null element 1
- C. Null  
Null
- D. A NullPointerException is thrown at runtime.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 43**

Given:

```
class Vehicle {  
    int x;  
    Vehicle(){  
        this(10); // line n1  
    }  
    Vehicle(int x) {  
        this.x = x;  
    }  
}  
  
class Car extends Vehicle {  
    int y;  
    Car() {  
        super();  
        this(20); // line n2  
    }  
    Car(int y) {  
        this.y = y;  
    }  
    public String toString() {  
        return super.x + ":" + this.y;  
    }  
}
```

And given the code fragment:

And given the code fragment:

```
Vehicle y = new Car();  
System.out.println(y);
```

What is the result?

- A. 10:20
- B. 0:20
- C. Compilation fails at line n1

D. Compilation fails at line n2

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 44**

Given the definitions of the MyString class and the Test class:

MyString.java:

```
package p1;
class MyString {
    String msg;
    MyString(String msg) {
        this.msg = msg;
    }
}
```

Test.java:

```
package p1;
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello " + new StringBuilder("Java SE 8"));
        System.out.println("Hello " + new MyString("Java SE 8"));
    }
}
```

What is the result?

- A) Hello Java SE 8  
Hello Java SE 8
- B) Hello java.lang.StringBuilder@<<hashcode1>>  
Hello pl.MyString@<<hashcode2>>
- C) Hello Java SE 8  
Hello pl.MyString@<<hashcode>>
- D) Compilation fails at the Test class.

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 45**

Given the code fragment:

```
public class Person {  
    String name;  
    int age = 25;  
  
    public Person(String name) {  
        this(); //line n1  
        setName(name);  
    }  
  
    public Person(String name, int age) {  
        Person(name); //line n2  
        setAge(age);  
    }  
  
    //setter and getter methods go here  
  
    public String show() {  
        return name + " " + age + " " + number ;  
    }  
    public static void main(String[] args) {  
        Person p1 = new Person("Jesse");  
        Person p2 = new Person("Walter", 52);  
        System.out.println(p1.show());  
        System.out.println(p2.show());  
    }  
}
```

What is the result?

- A. Jesse 25  
Walter 52
- B. Compilation fails only at line n1
- C. Compilation fails only at line n2
- D. Compilation fails at both line n1 and line n2

**Correct Answer:** B

Section: (none)  
Explanation

Explanation/Reference:

**QUESTION 46**

Given the following code for a Planet object:

```
public class Planet {  
    public String name;  
    public int moons;  
  
    public Planet(String name, int moons) {  
        this.name = name;  
        this.moons = moons;  
    }  
}
```

And the following main method:

```
public static void main(String[] args) {  
    Planet[] planets = {  
        new Planet("Mercury", 0),  
        new Planet("Venus", 0),  
        new Planet("Earth", 1),  
        new Planet("Mars", 2)  
    };  
  
    System.out.println(planets);  
    System.out.println(planets[2]);  
    System.out.println(planets[2].moons);  
}
```

What is the output?

- A) planets  
Earth  
1
- B) [LPlanets.Planet;@15db9742  
Earth  
1
- C) [LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
1
- D) [LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
[LPlanets.Moon;@7852e922
- E) [LPlanets.Planet;@15db9742  
Venus  
0

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### **QUESTION 47**

You are asked to develop a program for a shopping application, and you are given the following information:

- The application must contain the classes Toy, EduToy, and consToy. The Toy class is the superclass of the other two classes.
- The int calculatePrice (Toy t) method calculates the price of a toy. The void printToy (Toy t) method prints the details of a toy.

Which definition of the Toy class adds a valid layer of abstraction to the class hierarchy?

- A) public abstract class Toy{  
    public abstract int calculatePrice(Toy t);  
    public void printToy(Toy t) { /\* code goes here \*/ }  
}
- B) public abstract class Toy {  
    public int calculatePrice(Toy t) ;  
    public void printToy(Toy t) ;  
}
- C) public abstract class Toy {  
    public int calculatePrice(Toy t);  
    public final void printToy(Toy t){ /\* code goes here \*/ }  
}
- D) public abstract class Toy {  
    public abstract int calculatePrice(Toy t) { /\* code goes here \*/ }  
    public abstract void printToy(Toy t) { /\* code goes here \*/ }  
}

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 48

Given the following code:

```
int[] intArr = {15, 30, 45, 60, 75};  
intArr[2] = intArr[4];  
intArr[4] = 90;
```

What are the values of each element in intArr after this code has executed?

- A. 15, 60, 45, 90, 75
- B. 15, 90, 45, 90, 75
- C. 15, 30, 75, 60, 90
- D. 15, 30, 90, 60, 90
- E. 15, 4, 45, 60, 90

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 49**

Given the following array:

```
int[] intArr = {8, 16, 32, 64, 128};
```

Which two code fragments, independently, print each element in this array?

- A) 

```
for (int i : intArr) {
    System.out.print(intArr[i] + " ");
}
```
- B) 

```
for (int i : intArr) {
    System.out.print(i + " ");
}
```
- C) 

```
for (int i=0 : intArr) {
    System.out.print(intArr[i] + " ");
    i++;
}
```
- D) 

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(i + " ");
}
```
- E) 

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(intArr[i] + " ");
}
```
- F) 

```
for (int i; i < intArr.length; i++) {
    System.out.print(intArr[i] + " ");
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:****QUESTION 50**

Given the content of three files:

A.java:

```
public class A {  
    public void a() {}  
    int a;  
}
```

B.java:

```
public class B {  
    private int doStuff() {  
        private int x = 100;  
        return x++;  
    }  
}
```

C.java:

```
import java.io.*;  
package p1;  
class A {  
    public void main(String fileName) throws IOException {}  
}
```

Which statement is true?

Which statement is true?

- A. Only the A.java file compiles successfully.
- B. Only the B.java file compiles successfully.
- C. Only the C.java file compiles successfully.

- D. The A.java and B.java files compile successfully.
- E. The B.java and C.java files compile successfully.
- F. The A.java and C.java files compile successfully.

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 51**

Given the code fragment:

```
int[] array = {1, 2, 3, 4, 5};
```

And given the requirements:

- 1. Process all the elements of the array in the order of entry.
- 2. Process all the elements of the array in the reverse order of entry.
- 3. Process alternating elements of the array in the order of entry.

Which two statements are true?

- A. Requirements 1, 2, and 3 can be implemented by using the enhanced for loop.
- B. Requirements 1, 2, and 3 can be implemented by using the standard for loop.
- C. Requirements 2 and 3 CANNOT be implemented by using the standard for loop.
- D. Requirement 1 can be implemented by using the enhanced for loop.
- E. Requirement 3 CANNOT be implemented by using either the enhanced for loop or the standard for loop.

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 52**

Given:

```
public class TestScope {  
    public static void main(String[] args) {  
        int var1 = 200;  
        System.out.print(doCalc(var1));  
        System.out.print(" " + var1);  
    }  
    static int doCalc(int var1){  
        var1 = var1 * 2;  
        return var1;  
    }  
}
```

What is the result?

- A. 400 200
- B. 200 200
- C. 400 400
- D. Compilation fails.

**Correct Answer:** A

**Section:** (none)

**Explanation**

**Explanation/Reference:**

### QUESTION 53

Given the following class declarations:

- public abstract class Animal
- public interface Hunter
- public class Cat extends Animal implements Hunter

public class Tiger extends Cat

Which answer fails to compile?

- A) `ArrayList<Animal> myList = new ArrayList<>();  
myList.add(new Tiger());`
- B) `ArrayList<Hunter> myList = new ArrayList<>();  
myList.add(new Cat());`
- C) `ArrayList<Hunter> myList = new ArrayList<>();  
myList.add(new Tiger());`
- D) `ArrayList<Tiger> myList = new ArrayList<>();  
myList.add(new Cat());`
- E) `ArrayList<Animal> myList = new ArrayList<>();  
myList.add(new Cat());`

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Correct Answer:** E

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 54

Which statement is true about Java byte code?

- A. It can run on any platform.
- B. It can run on any platform only if it was compiled for that platform.
- C. It can run on any platform that has the Java Runtime Environment.
- D. It can run on any platform that has a Java compiler.
- E. It can run on any platform only if that platform has both the Java Runtime Environment and a Java compiler.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

Reference: <http://www.math.uni-hamburg.de/doc/java/tutorial/getStarted/intro/definition.html>

### QUESTION 55

Given:

```
public class MarkList {  
    int num;  
    public static void graceMarks(MarkList obj4) {  
        obj4.num += 10;  
    }  
    public static void main(String[] args) {  
        MarkList obj1 = new MarkList();  
        MarkList obj2 = obj1;  
        MarkList obj3 = null;  
        obj2.num = 60;  
        graceMarks(obj2);  
    }  
}
```

How many MarkList instances are created in memory at runtime?

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

**Correct Answer:** C

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 56**

Given:

```
public class Triangle {  
    static double area;  
    int b = 2, h = 3;  
    public static void main(String[] args) {  
        double p, b, h;          //line n1  
        if (area == 0) {  
            b = 3;  
            h = 4;  
            p = 0.5;  
        }  
        area = p * b * h;        //line n2  
        System.out.println("Area is " + area);  
    }  
}
```

What is the result?

- A. Area is 6.0
- B. Area is 3.0
- C. Compilation fails at line n1
- D. Compilation fails at line n2.

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**QUESTION 57**

Given the code fragment:

```
public class Test {  
    public static void main(String[] args) {  
        //line n1  
        switch (x) {  
            case 1:  
                System.out.println("One");  
                break;  
            case 2:  
                System.out.println("Two");  
                break;  
        }  
    }  
}
```

Which three code fragments can be independently inserted at line n1 to enable the code to print one?

- A. Byte x = 1;
- B. short x = 1;
- C. String x = "1";
- D. Long x = 1;
- E. Double x = 1;
- F. Integer x = new Integer ("1");

**Correct Answer:** D

**Section:** (none)

**Explanation**

**Explanation/Reference:**

#### QUESTION 58

Given:

```
public class App {  
  
    public static void main(String[] args) {  
        Boolean[] bool = new Boolean[2];  
  
        bool[0] = new Boolean(Boolean.parseBoolean("true"));  
        bool[1] = new Boolean(null);  
  
        System.out.println(bool[0] + " " + bool[1]);  
    }  
}
```

What is the result?

- A. True false
- B. True null
- C. Compilation fails
- D. A NullPointerException is thrown at runtime

**Correct Answer:** B

**Section:** (none)

**Explanation**

**Explanation/Reference:**

**NEW QUESTION 1**

Given:

```
public static void main(String[] args) {  
    String ta = "A ";  
    ta = ta.concat("B ");  
    String tb = "C ";  
    ta = ta.concat(tb);  
    ta.replace('C', 'D');  
    ta = ta.concat(tb);  
    System.out.println(ta);  
}
```

What is the result?

- A. A B C D
- B. A C D
- C. A C D D
- D. A B D
- E. A B D C

**Answer:** C**NEW QUESTION 2**

Given the code fragment:

```
public static void main(String[] args) {  
    int ans;  
    try {  
        int num = 10;  
        int div = 0;  
        ans = num / div;  
    } catch (ArithmaticException ae) {  
        ans = 0; // line n1  
    } catch (Exception e) {  
        System.out.println("Invalid calculation");  
    }  
    System.out.println("Answer = " + ans); // line n2  
}
```

What is the result?

- A. Answer = 0
- B. Invalid calculation
- C. Compilation fails only at line n1.
- D. Compilation fails only at line n2.
- E. Compilation fails at line n1 and line2.

**Answer:** C**Explanation:**

```
1  
2 public class Test {  
3     public static void main(String[] args) {  
4         int ans;  
5         try {  
6             int num = 10;  
7             int div = 0;  
8             ans = num / div;  
9         } catch (ArithmaticException ae) {  
10             ans = 0;  
11         } catch (Exception e) {  
12             System.out.println("Tnvalid calculation");  
13             variable ans might not have been initialized  
14         System.out.println("Answer = " + ans); //line n2  
15     }  
16 }
```

**NEW QUESTION 3**

Given the following classes:

```
public class Employee {
    public int salary;
}

public class Manager extends Employee {
    public int budget;
}

public class Director extends Manager {
    public int stockOptions;
}
```

And given the following main method:

```
public static void main(String[] args) {
    Employee employee = new Employee();
    Manager manager = new Manager();
    Director director = new Director();
    //line n1
}
```

Which two options fail to compile when placed at line n1 of the main method? (Choose two.)

- A. employee.salary = 50\_000;
- B. director.salary = 80\_000;
- C. employee.budget = 200\_000;
- D. manager.budget = 1\_000\_000;
- E. manager.stockOption = 500;
- F. director.stockOptions = 1\_000;

**Answer:** CE

#### NEW QUESTION 4

Given the code fragments:

Person.java:

```
public class Person {
    String name;
    int age;

    public Person(String n, int a) {
        name = n;
        age = a;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {
    for (Person p : list) {
        if (predicate.test(p)) {
            System.out.println(p.name + " ");
        }
    }
}

public static void main(String[] args) {
    List<Person> iList = Arrays.asList(new Person("Hank", 45),
                                         new Person("Charlie", 40),
                                         new Person("Smith", 38));
    //line n1
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

A

```
checkAge (iList, ( ) -> p. get Age ( ) > 40);
```

B

```
checkAge(iList, Person p -> p.getAge( ) > 40);
```

C

```
checkAge (iList, p -> p.getAge ( ) > 40);
```

D

```
checkAge(iList, (Person p) -> { p.getAge() > 40; });
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** C

#### NEW QUESTION 5

You are asked to develop a program for a shopping application, and you are given this information:

- The application must contain the classes Toy, EduToy, and ConsToy. The Toy class is the superclass of the other two classes.
- The int calculatePrice (Toy t) method calculates the price of a toy.
- The void printToy (Toy t) method prints the details of a toy.

Which definition of the Toy class adds a valid layer of abstraction to the class hierarchy?

A

```
public abstract class Toy{  
    public abstract int calculatePrice(Toy t);  
    public void printToy(Toy t) { /* code goes here */ }  
}
```

B

```
public abstract class Toy {  
    public int calculatePrice(Toy t) ;  
    public void printToy(Toy t) ;  
}
```

C

```
public abstract class Toy {  
    public int calculatePrice(Toy t);  
    public final void printToy(Toy t){ /* code goes here */ }  
}
```

D

```
public abstract class Toy {  
    public abstract int calculatePrice(Toy t) { /* code goes here */ }  
    public abstract void printToy(Toy t) { /* code goes here */ }  
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

#### NEW QUESTION 6

Given:

```
String stuff = "TV";
String res = null;

if (stuff.equals("TV")) {
    res = "Walter";
} else if (stuff.equals("Movie")) {
    res = "White";
} else {
    res = "No Result";
}
```

Which code fragment can replace the if block?

A  
stuff.equals ("TV") ? res= "Walter" : stuff.equals ("Movie") ?
res = "White" : res = "No Result";

B  
res = stuff.equals ("TV") ? "Walter" else stuff.equals
("Movie")? "White" : "No Result";

C  
res = stuff.equals ("TV") ? stuff.equals ("Movie")? "Walter" :
"White" : "No Result";

D  
res = stuff.equals ("TV")? "Walter" : stuff.equals ("Movie")?
"White" : "No Result";

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** D

#### NEW QUESTION 7

Given the code fragment:

```
LocalDate Time dt= LocalDateTime.of (2014, 7, 31, 1, 1);
dt.plusDays (30);
dt. plusMonths (1);
System.out.print (dt format (DateTimeFormatter. ISO_DATE) );
```

What is the result?

- A. An exception is thrown at runtime
- B. 07-31-2014
- C. 2014-07-31
- D. 2014-09-30

**Answer:** A

#### NEW QUESTION 8

Given this code for a Planet object:

```
public class Planet {  
    public String name;  
    public int moons;  
  
    public Planet(String name, int moons) {  
        this.name = name;  
        this.moons = moons;  
    }  
}
```

And this method:

```
public static void main(String[] args){  
    Planet[] planets = {  
        new Planet("Mercury", 0),  
        new Planet("Venus", 0),  
        new Planet("Earth", 1),  
        new Planet("Mars", 2)  
    };  
  
    System.out.println(planets);  
    System.out.println(planets[2].name);  
    System.out.println(planets[2].moons);  
}
```

What is the output?

- A  
planets  
Earth  
1
- B  
[LPlanets.Planet;@15db9742  
Earth  
1
- C  
[LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
1
- D  
[LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
[LPlanets.Moon;@7852e922
- E  
[LPlanets.Planet;@15db9742  
Venus  
0

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer: C**

#### NEW QUESTION 9

Given the code fragment:

```
public static void main(String[] args) {
    Short s1 = 200;
    Integer s2 = 400;
    Long s3 = (long) s1 + s2;           //line n1
    String s4 = (String) (s3 * s2);     //line n2
    System.out.println("Sum is " + s4);
}
```

What is the result?

- A. Sum is 600
- B. Compilation fails at line n1.
- C. Compilation fails at line n2.
- D. A ClassCastException is thrown at line n1.
- E. A ClassCastException is thrown at line n2.

**Answer:** C

#### NEW QUESTION 10

Given the code fragment:

```
public static void main(String[] args) {
    int data[] = {2010, 2013, 2014, 2015, 2014};
    int key = 2014;
    int count = 0;
    for (int e: data) {
        if (e != key) {
            continue;
            count++;
        }
    }
    System.out.print(count + " Found");
}
```

What is the result?

- A. Compilation fails.
- B. 0 Found
- C. 1 Found
- D. 3 Found

**Answer:** A

#### NEW QUESTION 10

Given:

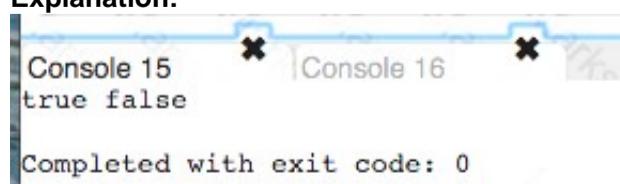
```
public class Test {
    public static void main(String[] args) {
        Test ts = new Test();
        System.out.print(isAvailable + " ");
        isAvailable= ts.doStuff();
        System.out.println(isAvailable);
    }
    public static boolean doStuff() {
        return !isAvailable;
    }
    static boolean isAvailable = true;
}
```

What is the result?

- A. Compilation fails.
- B. false true
- C. true false
- D. true true
- E. false false

**Answer:** C

#### Explanation:



```
Console 15      Console 16
true false
false true
Completed with exit code: 0
```

**NEW QUESTION 15**

Given:

```
class A {  
    public void test () {  
        System.out.println ("A");  
    }  
}  
class B extends A {  
    public void test () {  
        System.out.println ("B");  
    }  
}  
public class C extends A {  
    public void test () {  
        System.out.println ("C");  
    }  
  
    public static void main (String [] args) {  
        A b1 = new A ();  
        A b2 = new C ();  
  
        b1 = (A) b2;           //line n1  
        A b3 = (B) b2;         //line n2  
        b1.test ();  
        b3.test ();  
    }  
}
```

What is the result?

- A. AB
- B. AC
- C. CC
- D. A ClassCastException is thrown only at line n1.
- E. A ClassCastException is thrown only at line n2.

**Answer: B****NEW QUESTION 16**

Given the code fragment:

```
int n [] [] = {{1, 3}, {2, 4}};  
for (int i = n.length-1; i >= 0; i--) {  
    for (int y : n[i]) {  
        System.out.print (y);  
    }  
}
```

What is the result?

- A. 1324
- B. 2313
- C. 3142
- D. 4231

**Answer: D****NEW QUESTION 19**

Given:

```
public class Fieldinit {  
    char c;  
    boolean b;  
    float f;  
    void printAll() {  
        System.out.println ("c = " + c);  
        System.out.println ("b = " + b);  
        System.out.println ("f = " + f);  
    }  
    public static void main (String [] args) {  
        FieldInit f = new FieldInit ();  
        f.printAll ();  
    }  
}
```

What is the result?

A

```
c=  
b = false  
f = 0.0
```

B

```
c= null  
b = true  
f = 0.0
```

C

```
c=0  
b = false  
f = 0.0f
```

D

```
c= null  
b = false  
f = 0.0F
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

#### NEW QUESTION 22

Given:

```
class Patient {  
    String name;  
    public Patient (String name) {  
        this.name = name;  
    }  
}
```

And the code fragment:

```
8. public class Test {  
9.     public static void main (String [] args) {  
10.         List ps = new ArrayList ();  
11.         Patient p2 = new Patient ("Mike");  
12.         ps.add(p2);  
13.         // insert code here  
14.         if (f >= 0) {  
15.             System.out.print ("Mike Found");  
16.         }  
17.     }  
18. }
```

Which code fragment, when inserted at line 14, enables the code to print Mike Found?

A

```
int f = ps.indexOf (p2);
```

B

```
int f = ps.indexOf (Patient ("Mike") );
```

C

```
int f = ps.indexOf (new Patient "Mike") ;
```

D

```
Patient p = new Patient("Mike");  
int f = ps.indexOf(p)
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

#### NEW QUESTION 24

Given:

```
public class Test {  
    public static void main(String[] args) {  
        boolean a = new Boolean(Boolean.valueOf(args[0]));  
        boolean b = new Boolean(args[1]);  
        System.out.println(a + " " + b);  
    }  
}
```

And given the commands:

```
javac Test.java  
java Test 1 null
```

What is the result?

- A. 1 null
- B. true false
- C. false false
- D. true true
- E. A ClassCastException is thrown at runtime.

**Answer:** D

#### NEW QUESTION 28

Given:

```
public class MyClass {  
    public static void main(String[] args) {  
        String s = "Java SE 8 1";  
        int len = s.trim().length();  
        System.out.print(len);  
    }  
}
```

What is the result?

- A. Compilation fails.
- B. 11
- C. 8
- D. 9
- E. 10

**Answer:** B

#### NEW QUESTION 32

Given:

```
interface Readable {  
    public void readBook();  
    public void setBookMark();  
}  
  
abstract class Book implements Readable { // line n1  
    public void readBook() {}  
    // line n2  
}  
  
class EBook extends Book { // line n3  
    public void readBook() {}  
    // line n4  
}
```

And given the code fragment: Book book1 = new EBook(); book1.readBook();

Which option enables the code to compile?

- A) Replace the code fragment at line n1 with:  
class Book implements Readable {
- B) At line n2 insert:  
public abstract void setBookMark();
- C) Replace the code fragment at line n3 with:  
abstract class EBook extends Book {
- D) At line n4 insert:  
public void setBookMark() {}

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** D

#### NEW QUESTION 34

Given:

```
class Product {
    double price;
}

public class Test {
    public void updatePrice(Product product, double price) {
        price = price * 2;
        product.price = product.price + price;
    }
    public static void main(String[] args) {
        Product prt = new Product();
        prt.price = 200;
        double newPrice = 100;

        Test t = new Test();
        t.updatePrice(prt, newPrice);
        System.out.println(prt.price + " : " + newPrice);
    }
}
```

What is the result?

- A. 200.0 : 100.0
- B. 400.0 : 200.0
- C. 400.0 : 100.0
- D. Compilation fails.

**Answer:** C

#### NEW QUESTION 38

Which three statements are true about exception handling? (Choose three.)

- A. Only unchecked exceptions can be rethrown.
- B. All subclasses of the RuntimeException class are not recoverable.
- C. The parameter in a catch block is of Throwable type.
- D. All subclasses of the RuntimeException class must be caught or declared to be thrown.
- E. All subclasses of the RuntimeException class are unchecked exceptions.
- F. All subclasses of the Error class are not recoverable.

**Answer:** BCD

#### NEW QUESTION 39

Given:

```
class A {
    public void test() {
        System.out.println("A ");
    }
}

class B extends A {
    public void test() {
        System.out.println("B ");
    }
}

public class C extends A {
    public void test() {
        System.out.println("C ");
    }
}

public static void main(String[] args) {
    A b1 = new A();
    A b2 = new C();
    A b3 = (B) b2;           //line n1
    b1 = (A) b2;            //line n2
    b1.test();
    b3.test();
}
```

What is the result?

- A. AB
- B. AC

## QUESTION 41

- C. CC
- D. A ClassCastException is thrown only at line n1.
- E. A ClassCastException is thrown only at line n2.

**Answer:** D

## NEW QUESTION 42

Given:

```
interface I {  
    public void displayI();  
}  
abstract class C2 implements I {  
    public void displayC2() {  
        System.out.print("C2");  
    }  
}  
class C1 extends C2 {  
    public void displayI() {  
        System.out.print("C1");  
    }  
}
```

And the code fragment:

```
C2 obj1 = new C1();  
I obj2 = new C1();  
  
C2 s = (C2) obj2;  
I t = obj1;  
  
t.displayI();  
s.displayC2();
```

What is the result?

- A. C1C2
- B. C1C1
- C. Compilation fails.
- D. C2C2

**Answer:** A**Explanation:**

lund

src

App.java

x

+

```
1
2 interface I {
3     public void displayI();
4 }
5 abstract class C2 implements I {
6     public void displayC2() {
7         System.out.print("C2");
8     }
9 }
10 class C1 extends C2 {
11     public void displayI() {
12         System.out.print("C1");
13     }
14 }
15 }
16
17 public class App {
18     public static void main(String[] args) {
19         C2 obj1 = new C1();
20         I obj2 = new C1();
21
22         C2 s = (C2) obj2;
23         I t = obj1;
24
25         t.displayI();
26         s.displayC2();
27     }
28
29 }
```

Console 1    x    Console 2    x    Console 3    x    Console 4    x  
C1C2  
Completed with exit code: 0

**NEW QUESTION 43**

Given:

```
class Caller {  
    private void init () {  
        System.out.println("Initialized");  
    }  
  
    private void start () {  
        init();  
        System.out.println("Started");  
    }  
}  
  
public class TestCall {  
    public static void main(String[] args) {  
        Caller c = new Caller();  
        c.start(); // line n1  
        c.init(); // line n2  
    }  
}
```

What is the result?

- A. Compilation fails at line n1.
- B. InitializedStartedInitialized
- C. InitializedStarted
- D. Compilation fails at line n2.

**Answer:** D

#### NEW QUESTION 45

Given the code fragment:

```
public static void main(String[] args) {  
    StringBuilder sb = new StringBuilder("Java");  
    String s = "Java";  
  
    if (sb.toString().equals(s.toString())) {  
        System.out.println("Match 1");  
    } else if (sb.equals(s)) {  
        System.out.println("Match 2");  
    } else {  
        System.out.println("No Match");  
    }  
}
```

What is the result?

- A. Match 1
- B. Match 2
- C. No Match
- D. A NullPointerException is thrown at runtime.

**Answer:** A

#### NEW QUESTION 47

Given this class:

```
public class Rectangle {  
    private double length;  
    private double height;  
    private double area;  
  
    public void setLength(double length) {  
        this.length = length;  
    }  
    public void setHeight(double height) {  
        this.height = height;  
    }  
    public void setArea() {  
        area = length*height;  
    }  
}
```

Which two changes would encapsulate this class and ensure that the area field is always equal to length \* height whenever the Rectangle class is used?

- A. Call the setArea method at the end of the setHeight method.
- B. Call the setArea method at the beginning of the setHeight method.
- C. Call the setArea method at the end of the setLength method.
- D. Call the setArea method at the beginning of the setLength method.
- E. Change the setArea method to private.
- F. Change the area field to public.

**Answer:** AE

#### NEW QUESTION 51

Which statement is true about the switch statement?

- A. It must contain the default section.
- B. The break statement, at the end of each case block, is optional.
- C. Its case label literals can be changed at runtime.
- D. Its expression must evaluate to a collection of values.

**Answer:** B

#### NEW QUESTION 52

Given:

```
class Caller {  
    private void init () {  
        System.out.println("Initialized");  
    }  
  
    private void start () {  
        init();  
        System.out.println("Started");  
    }  
}  
  
public class TestCall {  
    public static void main(String[] args) {  
        Caller c = new Caller();  
        c.start();  
        c.init();  
    }  
}
```

What is the result?

- A. An exception is thrown at runtime.
- B. InitializedStartedInitialized
- C. InitializedStarted
- D. Compilation fails.

**Answer:** D

#### NEW QUESTION 55

Given the code fragment:

```
3. public static void main(String[] args) {  
4.     int x = 6;  
5.     while (isAvailable(x)) {  
6.         System.out.print(x);  
7.     }  
8. }  
10.  
11. public static boolean isAvailable(int x) {  
12.     return --x > 0 ? true : false;  
13. }
```

Which modification enables the code to print 54321?

- A. Replace line 6 with System.out.print (--x);
- B. At line 7, insert x--;
- C. Replace line 5 with while (is Available(--x)) {
- D. Replace line 12 with return (x > 0) ? false : true;

**Answer:** C

**NEW QUESTION 60**

Which statement best describes encapsulation?

- A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects.
- B. Encapsulation ensures that classes can be designed so that their methods are inheritable.
- C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.
- D. Encapsulation ensures that classes can be designed so that if a method has an argument MyType x, any subclass of MyType can be passed to that method.

**Answer:** A

**NEW QUESTION 64**

Which two statements are true? (Choose two.)

- A. Error class is unextendable.
- B. Error class is extendable.
- C. Error is a RuntimeException.
- D. Error is an Exception.
- E. Error is a Throwable.

**Answer:** BC

**NEW QUESTION 66**

Which three statements describe the object-oriented features of the Java language? (Choose three.)

- A. Objects cannot be reused.
- B. A subclass must override the methods from a superclass.
- C. Objects can share behaviors with other objects.
- D. A package must contain a main class.
- E. Object is the root class of all other objects.
- F. A main method must be declared in every class.

**Answer:** BCF

**NEW QUESTION 71**

Given the code fragment:

```
int nums1[] = {1, 2, 3};  
int nums2[] = {1, 2, 3, 4, 5};  
nums2 = nums1;  
for (int x : nums2){  
    System.out.print(x + ":" );  
}
```

What is the result?

- A. 1:2:3:4:5:
- B. 1:2:3:
- C. Compilation fails.
- D. An ArrayOutOfBoundsException is thrown at runtime.

**Answer:** A

**NEW QUESTION 75**

Which three statements are true about the structure of a Java class? (Choose three.)

- A. A class cannot have the same name as its field.
- B. A public class must have a main method.
- C. A class can have final static methods.
- D. A class can have overloaded private constructors.
- E. Fields need to be initialized before use.
- F. Methods and fields are optional components of a class.

**Answer:** BDE

**NEW QUESTION 76**

.....

# Exam Questions 1z0-808

Java SE 8 Programmer I

<https://www.2passeeasy.com/dumps/1z0-808/>



**NEW QUESTION 1**

Which one of the following code examples uses valid Java syntax?

- A.
- ```
public class Boat {  
  
    public static void main (String [] args) {  
        System.out.println ("I float.");  
    }  
}
```
- B.
- ```
public class Cake {  
    public static void main (String [] ) {  
        System.out.println ("Chocolate");  
    }  
}
```
- C.
- ```
public class Dog {  
    public void main (String [] args) {  
        System.out.println ("Squirrel.");  
    }  
}
```
- D.
- ```
public class Bank {  
    public static void main (String () args) {  
        System.out.println ("Earn interest.");  
    }  
}
```

- A. Option A  
B. Option B  
C. Option C  
D. Option D

**Answer:** A

**NEW QUESTION 2**

You are asked to create a method that accepts an array of integers and returns the highest value from that array.

Given the code fragment:

```
class Test{  
    public static void main(String[] args) {  
        int numbers[] = {12, 13, 42, 32, 15, 156, 23, 51, 12};  
        int[] keys = findMax(numbers);  
    }  
  
    /* line n1 */ {  
        int[] keys = new int[3];  
        /* code goes here*/  
        return keys;  
    }  
}
```

Which method signature do you use at line n1?

- A. public int findMax (int[] numbers)  
B. static int[] findMax (int[] max)  
C. static int findMax (int[] numbers)  
D. final int findMax (int[] )

Answer: C

**NEW QUESTION 3**

Given the code fragments:

Person.java:

```
public class Person {  
    String name;  
    int age;  
  
    public Person(String n, int a) {  
        name = n;  
        age = a;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getAge() {  
        return age;  
    }  
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {  
    for (Person p : list) {  
        if (predicate.test(p)) {  
            System.out.println(p.name + " ");  
        }  
    }  
}  
  
public static void main(String[] args) {  
    List<Person> iList = Arrays.asList(new Person("Hank", 45),  
                                         new Person("Charlie", 40),  
                                         new Person("Smith", 38));  
    //line n1  
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

- A  
checkAge (iList, ( ) -> p. get Age ( ) > 40);
- B  
checkAge(iList, Person p -> p.getAge( ) > 40);
- C  
checkAge (iList, p -> p.getAge ( ) > 40);
- D  
checkAge(iList, (Person p) -> { p.getAge() > 40; });

- A. Option A  
B. Option B  
C. Option C  
D. Option D

Answer: C

**NEW QUESTION 4**

You are asked to develop a program for a shopping application, and you are given this information:

- The application must contain the classes Toy, EduToy, and ConsToy. The Toy class is the superclass of the other two classes.
- The int calculatePrice (Toy t) method calculates the price of a toy.
- The void printToy (Toy t) method prints the details of a toy.

Which definition of the Toy class adds a valid layer of abstraction to the class hierarchy?

- A
- ```
public abstract class Toy{
    public abstract int calculatePrice(Toy t);
    public void printToy(Toy t) { /* code goes here */ }
}
```
- B
- ```
public abstract class Toy {
    public int calculatePrice(Toy t) ;
    public void printToy(Toy t) ;
}
```
- C
- ```
public abstract class Toy {
    public int calculatePrice(Toy t);
    public final void printToy(Toy t){ /* code goes here */ }
}
```
- D
- ```
public abstract class Toy {
    public abstract int calculatePrice(Toy t) { /* code goes here */ }
    public abstract void printToy(Toy t) { /* code goes here */ }
}
```

- A. Option A  
B. Option B  
C. Option C  
D. Option D

**Answer:** A

#### NEW QUESTION 5

Given the definitions of the MyString class and the Test class:

```
package p1;
class MyString {
    String msg;
    MyString(String msg) {
        this.msg = msg;
    }
}
```

Test.java:

```
package p1;
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello " + new StringBuilder("Java SE 8"));
        System.out.println("Hello " + new MyString("Java SE 8").msg);
    }
}
```

What is the result?

- A
- ```
Hello Java SE 8
Hello Java SE 8
```
- B
- ```
Hello java.lang.StringBuilder@<<hashcode1>>
Hello p1.MyString@<<hashcode2>>
```
- C
- ```
Hello Java SE 8
Hello p1.MyString@<<hashcode>>
```
- D Compilation fails at the Test class

- A. Option A

- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** D

#### NEW QUESTION 6

Given this code for a Planet object:

```
public class Planet {  
    public String name;  
    public int moons;  
  
    public Planet(String name, int moons) {  
        this.name = name;  
        this.moons = moons;  
    }  
}
```

And this method:

```
public static void main(String[] args){  
    Planet[] planets = {  
        new Planet("Mercury", 0),  
        new Planet("Venus", 0),  
        new Planet("Earth", 1),  
        new Planet("Mars", 2)  
    };  
  
    System.out.println(planets);  
    System.out.println(planets[2].name);  
    System.out.println(planets[2].moons);  
}
```

What is the output?

- A  
planets  
Earth  
1
- B  
[LPlanets.Planet;@15db9742  
Earth  
1
- C  
[LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
1
- D  
[LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
[LPlanets.Moon;@7852e922
- E  
[LPlanets.Planet;@15db9742  
Venus  
0

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** C

**NEW QUESTION 7**

Given:

```
public class App {  
    int count;  
    public static void displayMsg () {  
        count++; // line n1  
        System.out.println ("Welcome +" "Visit Count: "+count); // line n2  
    }  
    public static void main (String [] args) {  
        App.displayMsg (); // line n3  
        App.displayMsg (); // line n4  
    }  
}
```

What is the result?

- A. Compilation fails at line n3 and line n4.
- B. Compilation fails at line n1 and line n2.
- C. Welcome Visit Count:1Welcome Visit Count: 1
- D. Welcome Visit Count:1Welcome Visit Count: 2

**Answer:** B**NEW QUESTION 8**

Which two are benefits of polymorphism? (Choose two.)

- A. Faster code at runtime
- B. More efficient code at runtime
- C. More dynamic code at runtime
- D. More flexible and reusable code
- E. Code that is protected from extension by other classes

**Answer:** BD**NEW QUESTION 9**

Which statement is true about the switch statement?

- A. It must contain the default section.
- B. The break statement, at the end of each case block, is mandatory.
- C. Its case label literals can be changed at runtime.
- D. Its expression must evaluate to a single value.

**Answer:** D**NEW QUESTION 10**

Given the code from the Greeting.Java file:

```
public class Greeting {  
    public static void main(String[] args) {  
        System.out.println("Hello " + args[0]);  
    }  
}
```

Which set of commands prints Hello Duke in the console?

- A) javac Greeting  
java Greeting Duke
- B) javac Greeting.java Duke  
java Greeting
- C) javac Greeting.java  
java Greeting Duke
- D) javac Greeting.java  
java Greeting.class Duke

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** C

**NEW QUESTION 10**

Which two statements are true about Java byte code? (Choose two.)

- A. It can be serialized across network.
- B. It can run on any platform that has a Java compiler.
- C. It can run on any platform.
- D. It has ".java" extension.
- E. It can run on any platform that has the Java Runtime Environment.

**Answer:** AE

**NEW QUESTION 13**

Given:

```
public class Fieldinit {  
    char c;  
    boolean b;  
    float f;  
    void printAll() {  
        System.out.println ("c = " + c);  
        System.out.println ("b = " + b);  
        System.out.println ("f = " + f);  
    }  
    public static void main (String [] args) {  
        FieldInit f = new FieldInit ();  
        f.printAll ();  
    }  
}
```

What is the result?

**A**

```
c=  
b = false  
f = 0.0
```

**B**

```
c= null  
b = true  
f = 0.0
```

**C**

```
c=0  
b = false  
f = 0.0f
```

**D**

```
c= null  
b = false  
f = 0.0F
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

**NEW QUESTION 16**

Given:

```
class Patient {  
    String name;  
    public Patient (String name) {  
        this.name = name;  
    }  
}
```

And the code fragment:

```
8. public class Test {  
9.     public static void main (String [] args) {  
10.         List ps = new ArrayList ();  
11.         Patient p2 = new Patient ("Mike");  
12.         ps.add(p2);  
13.  
14.         // insert code here  
15.  
16.         if (f >= 0) {  
17.             System.out.print ("Mike Found");  
18.         }  
19.     }  
20. }
```

Which code fragment, when inserted at line 14, enables the code to print Mike Found?

A

```
int f = ps.indexOf (p2);
```

B

```
int f = ps.indexOf (Patient ("Mike") );
```

C

```
int f = ps.indexOf (new Patient "Mike") );
```

D

```
Patient p = new Patient("Mike");  
int f = ps.indexOf(p)
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer: A**

#### NEW QUESTION 20

Given:

```
public class MyClass {  
    public static void main(String[] args) {  
        String s = "Java SE 8 1";  
        int len = s.trim().length();  
        System.out.print(len);  
    }  
}
```

What is the result?

- A. Compilation fails.
- B. 11
- C. 8
- D. 9
- E. 10

Answer: B

**NEW QUESTION 24**

Given:

```
class Product {  
    double price;  
}  
  
public class Test {  
    public void updatePrice(Product product, double price) {  
        price = price * 2;  
        product.price = product.price + price;  
    }  
    public static void main(String[] args) {  
        Product prt = new Product();  
        prt.price = 200;  
        double newPrice = 100;  
  
        Test t = new Test();  
        t.updatePrice(prt, newPrice);  
        System.out.println(prt.price + " : " + newPrice);  
    }  
}
```

What is the result?

- A. 200.0 : 100.0
- B. 400.0 : 200.0
- C. 400.0 : 100.0
- D. Compilation fails.

Answer: C

**NEW QUESTION 25**

Given:

```
class X {  
    static int i;  
    int j;  
    public static void main(String[] args) {  
        X x1 = new X();  
        X x2 = new X();  
        x1.i = 3;  
        x1.j = 4;  
        x2.i = 5;  
        x2.j = 6;  
        System.out.println(  
            x1.i + " " +  
            x1.j + " " +  
            x2.i + " " +  
            x2.j);  
    }  
}
```

What is the result?

- A. 3 4 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 4 6

Answer: C

**NEW QUESTION 27**

Given the code fragment:

```
public static void main(String[] args) {  
    LocalDate date = LocalDate.of(2012, 01, 32);  
    date.plusDays(10);  
    System.out.println(date);  
}
```

What is the result?

- A. 2012-02-10

- B. 2012-02-11
- C. Compilation fails
- D. A DateTimeException is thrown at runtime.

**Answer:** D

#### NEW QUESTION 29

Given the code fragment:

```
abstract class Planet {  
    protected void revolve() { //line n1  
    }  
  
    abstract void rotate(); //line n2  
}  
  
class Earth extends Planet {  
    void revolve() { //line n3  
    }  
  
    protected void rotate() { //line n4  
    }  
}
```

Which two modifications, made independently, enable the code to compile? (Choose two.)

- A. Make the method at line n1 public.
- B. Make the method at line n2 public.
- C. Make the method at line n3 public.
- D. Make the method at line n3 protected.
- E. Make the method at line n4 public.

**Answer:** CD

#### NEW QUESTION 31

Given this class:

```
public class CheckingAccount {  
    public int amount;  
    //line n1  
}
```

And given this main method, located in another class:

```
public static void main(String[] args) {  
    CheckingAccount acct = new CheckingAccount();  
    //line n2  
}
```

Which three pieces of code, when inserted independently, set the value of amount to 100?

A

At line n1 insert:

```
public CheckingAccount() {
    amount = 100;
}
```

B

At line n2 insert:

```
this.amount = 100;
```

C

At line n2 insert:

```
amount = 100;
```

D

At line n1 insert:

```
public CheckingAccount() {
    this.amount = 100;
}
```

E

At line n2 insert:

```
acct.amount = 100;
```

F

At line n1 insert:

```
public CheckingAccount() {
    acct.amount = 100;
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

**Answer:** DE

### NEW QUESTION 32

Given the code fragment:

```
7. StringBuilder sb1 = new StringBuilder("Duke");
8. String str1 = sb1.toString();
9. // insert code here
10. System.out.print(str1 == str2);
```

Which code fragment, when inserted at line 9, enables the code to print true?

- A. String str2 = str1;
- B. String str2 = new String(str1);
- C. String str2 = sb1.toString();
- D. String str2 = "Duke";

**Answer:** A

### NEW QUESTION 33

Given the code fragment:

```
public static void main(String[] args) {
    LocalDate date = LocalDate.of(2012, 1, 30);
    date.plusDays(10);
    System.out.println(date);
}
```

What is the result?

- A. 2012-02-10
- B. 2012-01-30
- C. 2012-02-10 00:00
- D. A DateTimeException is thrown at runtime.

Answer: C

**NEW QUESTION 38**

Which two code fragments cause a compilation error? (Choose two.)

- A. float flt = 100.00F;
- B. float flt = (float) 1\_11.00;
- C. Float flt = 100.00;
- D. double y1 = 203.22;float flt = y1;
- E. int y2 = 100;float flt = (float) y2 ;

Answer: AD

**NEW QUESTION 42**

What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?

- A. Encapsulation
- B. Inheritance
- C. Abstraction
- D. Instantiation
- E. Polymorphism

Answer: A

**Explanation:**

Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

**NEW QUESTION 47**

Given the code fragment:

```
int wd = 0;
String days[] = {"sun", "mon", "wed", "sat"};
for (String s:days) {
    switch (s) {
        case "sat":
        case "sun":
            wd -= 1;
            break;
        case "mon":
            wd++;
        case "wed":
            wd += 2;
    }
}
System.out.println(wd);
```

What is the result?

- A. 3
- B. 4
- C. -1
- D. Compilation fails.

Answer: A

**NEW QUESTION 51**

Given:

```
public class Test {  
    int x, y;  
  
    public Test(int x, int y) {  
        initialize(x, y);  
    }  
  
    public void initialize(int x, int y) {  
        this.x = x * x;  
        this.y = y * y;  
    }  
  
    public static void main(String[] args) {  
        int x = 3, y = 5;  
        Test obj = new Test(x, y);  
        System.out.println(x + " " + y);  
    }  
}
```

What is the result?

- A. Compilation fails.
- B. 3 5
- C. 0 0
- D. 9 25

**Answer:** B

#### NEW QUESTION 55

Which three are advantages of the Java exception mechanism? (Choose three.)

- A. Improves the program structure because the error handling code is separated from the normal program function
- B. Provides a set of standard exceptions that covers all possible errors
- C. Improves the program structure because the programmer can choose where to handle exceptions
- D. Improves the program structure because exceptions must be handled in the method in which they occurred
- E. Allows the creation of new exceptions that are customized to the particular program being created

**Answer:** ACE

#### NEW QUESTION 58

Given this class:

```
public class Rectangle {  
    private double length;  
    private double height;  
    private double area;  
  
    public void setLength(double length) {  
        this.length = length;  
    }  
    public void setHeight(double height) {  
        this.height = height;  
    }  
    public void setArea() {  
        area = length*height;  
    }  
}
```

Which two changes would encapsulate this class and ensure that the area field is always equal to length \* height whenever the Rectangle class is used?

- A. Call the setArea method at the end of the setHeight method.
- B. Call the setArea method at the beginning of the setHeight method.
- C. Call the setArea method at the end of the setLength method.
- D. Call the setArea method at the beginning of the setLength method.
- E. Change the setArea method to private.
- F. Change the area field to public.

**Answer:** AE

#### NEW QUESTION 61

Given:

```
class Caller {  
    private void init () {  
        System.out.println("Initialized");  
    }  
  
    private void start () {  
        init();  
        System.out.println("Started");  
    }  
}  
  
public class TestCall {  
    public static void main(String[] args) {  
        Caller c = new Caller();  
        c.start();  
        c.init();  
    }  
}
```

What is the result?

- A. An exception is thrown at runtime.
- B. InitializedStartedInitialized
- C. InitializedStarted
- D. Compilation fails.

**Answer:** D

#### NEW QUESTION 63

Given this segment of code:

```
ArrayList<Cycle> myList = new ArrayList<>();  
myList.add(new MotorCycle());
```

Which two statements, if either were true, would make the code compile? (Choose two.)

- A. MotorCycle is an interface that implements the Cycle class.
- B. Cycle is an interface that is implemented by the MotorCycle class.
- C. Cycle is an abstract superclass of MotorCycle.
- D. Cycle and MotorCycle both extend the Transportation superclass.
- E. Cycle and MotorCycle both implement the Transportation interface.
- F. MotorCycle is a superclass of Cycle.

**Answer:** BC

#### NEW QUESTION 64

Which two statements are true? (Choose two.)

- A. Error class is unextendable.
- B. Error class is extendable.
- C. Error is a RuntimeException.
- D. Error is an Exception.
- E. Error is a Throwable.

**Answer:** BC

#### NEW QUESTION 69

Which three statements describe the object-oriented features of the Java language? (Choose three.)

- A. Objects cannot be reused.
- B. A subclass must override the methods from a superclass.
- C. Objects can share behaviors with other objects.
- D. A package must contain a main class.
- E. Object is the root class of all other objects.
- F. A main method must be declared in every class.

**Answer:** BCF

#### NEW QUESTION 74

Which statement will empty the contents of a StringBuilder variable named sb?

- A. s
- B. deleteAll();
- C. s
- D. delete (0, s)

- E. size () ;
- F. s
- G. delete (0, s
- H. length () ;
- I. s
- J. removeAll () ;

**Answer:** C

#### NEW QUESTION 75

Given the code fragment:

```
String[] strs = {"A", "B"};
int idx = 0;
for (String s : strs) {
    strs[idx].concat(" element " + idx);
    idx++;
}
for (idx = 0; idx < strs.length; idx++) {
    System.out.println(strs[idx]);
}
```

What is the result?

- A. AB
- B. A element 0B element 1
- C. A NullPointerException is thrown at runtime.
- D. A 0B 1

**Answer:** C

#### NEW QUESTION 76

Which three statements are true about the structure of a Java class? (Choose three.)

- A. A public class must have a main method.
- B. A class can have only one private constructors.
- C. A method can have the same name as a field.
- D. A class can have overloaded static methods.
- E. The methods are mandatory components of a class.
- F. The fields need not be initialized before use.

**Answer:** ACE

#### NEW QUESTION 78

Which three statements are true about the structure of a Java class? (Choose three.)

- A. A class cannot have the same name as its field.
- B. A public class must have a main method.
- C. A class can have final static methods.
- D. A class can have overloaded private constructors.
- E. Fields need to be initialized before use.
- F. Methods and fields are optional components of a class.

**Answer:** BDE

#### NEW QUESTION 82

Given:

```
public class App {
    public static void main(String[] args) {
        int i = 10;
        int j = 20;
        int k = (j += i) / 5;
        System.out.print(i + " : " + j + " : " + k);
    }
}
```

What is the result?

- A. 10 : 30 : 6
- B. 10 : 22 : 22
- C. 10 : 22 : 20
- D. 10 : 22 : 6

**Answer:** A

#### NEW QUESTION 84

.....

## THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual 1z0-808 Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the 1z0-808 Product From:

<https://www.2passey.com/dumps/1z0-808/>

## Money Back Guarantee

### 1z0-808 Practice Exam Features:

- \* 1z0-808 Questions and Answers Updated Frequently
- \* 1z0-808 Practice Questions Verified by Expert Senior Certified Staff
- \* 1z0-808 Most Realistic Questions that Guarantee you a Pass on Your First Try
- \* 1z0-808 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

# Exam Questions 1z0-808

Java SE 8 Programmer I

<https://www.2passeeasy.com/dumps/1z0-808/>



**NEW QUESTION 1**

Given the code fragment:

```
public static void main(String[] args) {  
    int ans;  
    try {  
        int num = 10;  
        int div = 0;  
        ans = num / div;  
    } catch (ArithmetricException ae) {  
        ans = 0; // line n1  
    } catch (Exception e) {  
        System.out.println("Invalid calculation");  
    }  
    System.out.println("Answer = " + ans); // line n2  
}
```

What is the result?

- A. Answer = 0
- B. Invalid calculation
- C. Compilation fails only at line n1.
- D. Compilation fails only at line n2.
- E. Compilation fails at line n1 and line2.

**Answer:** C

**Explanation:**

```
1  
2 public class Test {  
3     public static void main(String[] args) {  
4         int ans;  
5         try {  
6             int num = 10;  
7             int div = 0;  
8             ans = num / div;  
9         } catch (ArithmetricException ae) {  
10            ans = 0;  
11        } catch (Exception e) {  
12            System.out.println("Invalid calculation");  
13            ✖ variable ans might not have been initialized  
14            System.out.println("Answer = " + ans); //line n2  
15        }  
16    }  
17 }
```

**NEW QUESTION 2**

Given the code fragments:

Person.java:

```
public class Person {  
    String name;  
    int age;  
  
    public Person(String n, int a) {  
        name = n;  
        age = a;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getAge() {  
        return age;  
    }  
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {  
    for (Person p : list) {  
        if (predicate.test(p)) {  
            System.out.println(p.name + " ");  
        }  
    }  
}  
  
public static void main(String[] args) {  
    List<Person> iList = Arrays.asList(new Person("Hank", 45),  
   new Person("Charlie", 40),  
   new Person("Smith", 38));  
    //line n1  
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

- A  
checkAge (iList, ( ) -> p. get Age ( ) > 40);
- B  
checkAge(iList, Person p -> p.getAge( ) > 40);
- C  
checkAge (iList, p -> p.getAge ( ) > 40);
- D  
checkAge(iList, (Person p) -> { p.getAge() > 40; });

- A. Option A  
B. Option B  
C. Option C  
D. Option D

Answer: C

#### NEW QUESTION 3

Given:

```
String stuff = "TV";
String res = null;

if (stuff.equals("TV")) {
    res = "Walter";
} else if (stuff.equals("Movie")) {
    res = "White";
} else {
    res = "No Result";
}
```

Which code fragment can replace the if block?

A

```
stuff.equals ("TV") ? res= "Walter" : stuff.equals ("Movie") ?
res = "White" : res = "No Result";
```

B

```
res = stuff.equals ("TV") ? "Walter" else stuff.equals
("Movie")? "White" : "No Result";
```

C

```
res = stuff.equals ("TV") ? stuff.equals ("Movie")? "Walter" :
"White" : "No Result";
```

D

```
res = stuff.equals ("TV")? "Walter" : stuff.equals ("Movie")?
"White" : "No Result";
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** D

#### NEW QUESTION 4

Which two are benefits of polymorphism? (Choose two.)

- A. Faster code at runtime
- B. More efficient code at runtime
- C. More dynamic code at runtime
- D. More flexible and reusable code
- E. Code that is protected from extension by other classes

**Answer:** BD

#### NEW QUESTION 5

Given:

```
public class Test {
    public static void main(String[] args) {
        boolean a = new Boolean(Boolean.valueOf(args[0]));
        boolean b = new Boolean(args[1]);
        System.out.println(a + " " + b);
    }
}
```

And given the commands:

```
javac Test.java
java Test 1 null
```

What is the result?

- A. 1 null
- B. true false
- C. false false
- D. true true
- E. A ClassCastException is thrown at runtime.

**Answer:** D

**NEW QUESTION 6**

Given the code fragment:

```
public class Employee {  
    String name;  
    boolean contract;  
    double salary;  
    Employee() {  
        // line n1  
    }  
    public String toString(){  
        return name + ":" + contract + ":" + salary;  
    }  
    public static void main(String[] args) {  
        Employee e = new Employee();  
        // line n2  
        System.out.print(e);  
    }  
}
```

Which two modifications, when made independently, enable the code to print Joe:true: 100.0? (Choose two.)

 A) Replace line n2 with:

```
e.name = "Joe";  
e.contract = true;  
e.salary = 100;
```

 B) Replace line n2 with:

```
this.name = "Joe";  
this.contract = true;  
this.salary = 100;
```

 C) Replace line n1 with:

```
this.name = new String("Joe");  
this.contract = new Boolean(true);  
this.salary = new Double(100);
```

 D) Replace line n1 with:

```
name = "Joe";  
contract = TRUE;  
salary = 100.0f;
```

 E) Replace line n1 with:

```
this("Joe", true, 100);
```

A. Option A

B. Option B

C. Option C

D. Option D

E. Option E

**Answer:** AC**NEW QUESTION 7**

Given:

```
class Product {  
    double price;  
}  
  
public class Test {  
    public void updatePrice(Product product, double price) {  
        price = price * 2;  
        product.price = product.price + price;  
    }  
    public static void main(String[] args) {  
        Product prt = new Product();  
        prt.price = 200;  
        double newPrice = 100;  
  
        Test t = new Test();  
        t.updatePrice(prt, newPrice);  
        System.out.println(prt.price + " : " + newPrice);  
    }  
}
```

What is the result?

- A. 200.0 : 100.0
- B. 400.0 : 200.0
- C. 400.0 : 100.0
- D. Compilation fails.

**Answer:** C

#### NEW QUESTION 8

Given:

```
class X {  
    static int i;  
    int j;  
    public static void main(String[] args) {  
        X x1 = new X();  
        X x2 = new X();  
        x1.i = 3;  
        x1.j = 4;  
        x2.i = 5;  
        x2.j = 6;  
        System.out.println(  
            x1.i + " " +  
            x1.j + " " +  
            x2.i + " " +  
            x2.j);  
    }  
}
```

What is the result?

- A. 3 4 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 4 6

**Answer:** C

#### NEW QUESTION 9

Which is true about the switch statement?

- A. Its expression can evaluate to a collection of values.
- B. The break statement, at the end of each case block, is optional.
- C. Its case label literals can be changed at runtime.
- D. It must contain the default section.

**Answer:** B

#### NEW QUESTION 10

Given:

```
class X {  
    int i;  
    static int j;  
    public static void main(String[] args) {  
        X x1 = new X();  
        X x2 = new X();  
        x1.i = 3;  
        x1.j = 4;  
        x2.i = 5;  
        x2.j = 6;  
        System.out.println(  
            x1.i + " " +  
            x1.j + " " +  
            x2.i + " " +  
            x2.j);  
    }  
}
```

What is the result?

- A. 3 4 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 5 6

**Answer:** D

**Explanation:**

```
3 6 5 6
Completed with exit code: 0
```

**NEW QUESTION 10**

Given:

```
class Student {
    String name;
    public Student(String name) {
        this.name = name;
    }
}

public class Test {
    public static void main(String[] args) {
        Student[] students = new Student[3];
        students[1] = new Student("Richard");
        students[2] = new Student("Donald");
        for (Student s : students) {
            System.out.println(" " + s.name);
        }
    }
}
```

What is the result?

- A. nullRichardDonald
- B. RichardDonald
- C. Compilation fails.
- D. An ArrayIndexOutOfBoundsException is thrown at runtime.
- E. A NullPointerException is thrown at runtime.

**Answer:** E**NEW QUESTION 14**

Which three are advantages of the Java exception mechanism? (Choose three.)

- A. Improves the program structure because the error handling code is separated from the normal program function
- B. Provides a set of standard exceptions that covers all possible errors
- C. Improves the program structure because the programmer can choose where to handle exceptions
- D. Improves the program structure because exceptions must be handled in the method in which they occurred
- E. Allows the creation of new exceptions that are customized to the particular program being created

**Answer:** ACE**NEW QUESTION 15**

Given:

```
class Caller {
    private void init () {
        System.out.println("Initialized");
    }

    private void start () {
        init();
        System.out.println("Started");
    }
}

public class TestCall {
    public static void main(String[] args) {
        Caller c = new Caller();
        c.start();
        c.init();
    }
}
```

What is the result?

- A. An exception is thrown at runtime.

- B. InitializedStartedInitialized
- C. InitializedStarted
- D. Compilation fails.

**Answer:** D

#### NEW QUESTION 20

Which statement best describes encapsulation?

- A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects.
- B. Encapsulation ensures that classes can be designed so that their methods are inheritable.
- C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.
- D. Encapsulation ensures that classes can be designed so that if a method has an argument MyType x, any subclass of MyType can be passed to that method.

**Answer:** A

#### NEW QUESTION 21

Which three statements describe the object-oriented features of the Java language? (Choose three.)

- A. Objects cannot be reused.
- B. A subclass must override the methods from a superclass.
- C. Objects can share behaviors with other objects.
- D. A package must contain a main class.
- E. Object is the root class of all other objects.
- F. A main method must be declared in every class.

**Answer:** BCF

#### NEW QUESTION 22

Given the code fragment:

```
int nums1[] = {1, 2, 3};  
int nums2[] = {1, 2, 3, 4, 5};  
nums 2 = nums 1;  
for (int x : nums2){  
    System.out.print(x + ":" );  
}
```

What is the result?

- A. 1:2:3:4:5:
- B. 1:2:3:
- C. Compilation fails.
- D. An ArrayOutOfBoundsException is thrown at runtime.

**Answer:** A

#### NEW QUESTION 26

Which three statements are true about the structure of a Java class? (Choose three.)

- A. A public class must have a main method.
- B. A class can have only one private constructors.
- C. A method can have the same name as a field.
- D. A class can have overloaded static methods.
- E. The methods are mandatory components of a class.
- F. The fields need not be initialized before use.

**Answer:** ACE

#### NEW QUESTION 31

.....

## THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual 1z0-808 Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the 1z0-808 Product From:

<https://www.2passey.com/dumps/1z0-808/>

## Money Back Guarantee

### 1z0-808 Practice Exam Features:

- \* 1z0-808 Questions and Answers Updated Frequently
- \* 1z0-808 Practice Questions Verified by Expert Senior Certified Staff
- \* 1z0-808 Most Realistic Questions that Guarantee you a Pass on Your First Try
- \* 1z0-808 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

➤ **Vendor: Oracle**

➤ **Exam Code: 1Z0-808**

➤ **Exam Name: Java SE 8 Programmer I**

➤ **New Questions (July/2018)**

Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps

**NEW QUESTION 290**

Given:

```
class C2 {  
    public void displayC2() {  
        System.out.print("C2");  
    }  
}  
interface I {  
    public void displayI();  
}  
class C1 extends C2 implements I {  
    public void displayI() {  
        System.out.print("C1");  
    }  
}
```

And given the code fragment:

```
C2 obj1 = new C1();  
I obj2 = new C1();  
  
C2 s = obj2;  
I t = obj1;  
  
t.displayI();  
s.displayC2()
```

What is the result?

- A. C2C2
- B. C1C2
- C. C1C1
- D. Compilation fails

**Answer: A**

**NEW QUESTION 291**

Given:

```
package clothing;
public class Shirt {
    public static String getColor() {
        return "Green";
    }
}
```

Given the code fragment:

```
package clothing.pants;
// line n1
public class Jeans {
    public void matchShirt(){
        //line n2
        if(color.equals("Green")) {
            System.out.print("Fit")
        }
    }
    public static void main (String[] args) {
        Jeans trouser = new Jeans();
        trouser.matchShirt();
    }
}
```

Which two sets of actions, independently, enable the code fragment to print Fit?

- A. At line n1 insert:import clothing.Shirt;At line n2 insert:String color = getColor();
- B. At line n1 insert:import clothing.\*;At line n2 insert:String color = Shirt.getColor();
- C. At line n1 insert:import static clothing.Shirt.getColor();At line n2 insert:String color = getColor();
- D. At line n1 no changes required.At line n2 insert:String color = Shirt.getColor();
- E. At line n1 insert:import clothing;At line n2 insert:String color = Shirt.getColor();

**Answer: A**

**NEW QUESTION 292**

Given the code fragments:

```
class Student {  
    String name;  
    int age;  
}
```

And,

```
4. public class Test {  
5.     public static void main(String[] args) {  
6.         Student s1 = new Student();  
7.         Student s2 = new Student();  
8.         Student s3 = new Student();  
9.         s1 = s3;  
10.        s3 = s2;  
11.        s2 = null;  
12.    }  
13.}
```

Which statement is true?

- A. After line 11, three objects are eligible for garbage collection.
- B. After line 11, two objects are eligible for garbage collection.
- C. After line 11, one object is eligible for garbage collection.
- D. After line 11, none of the objects are eligible for garbage collection.

**Answer: C**

#### **NEW QUESTION 293**

Given the code fragment:

```
int wd = 0;  
String days[] = {"sun", "mon", "wed", "sat");  
for (String s:days) {  
    switch (s) {  
        case "sat":  
        case "sun":  
            wd -= 1;  
            break;  
        case "mon":  
            wd++;  
        case "wed":  
            wd += 2;  
    }  
}  
System.out.println(wd);
```

What is the result?

- A. 3
- B. 4
- C. -1
- D. Compilation fails

**Answer: B**

**NEW QUESTION 294**

Given the code fragment:

```
public static void main(String[] args) {  
    LocalDate date = LocalDate.of(2012, 01, 32);  
    date.plusDays(10);  
    System.out.println(date);  
}
```

What is the result?

- A. 2012-02-10
- B. 2012-02-11
- C. Compilation fails
- D. A DateTimeException is thrown at runtime

**Answer: C**

**NEW QUESTION 295**

Given:

```
public class App {  
    public static void main(String[] args) {  
        int i = 10;  
        int j = 20;  
        int k = j += i / 5;  
        System.out.print(i + " : " + j + " : " + k);  
    }  
}
```

What is the result?

- A. 10 : 30 : 6
- B. 10 : 22 : 22
- C. 10 : 22 : 20
- D. 10 : 22 : 6

**Answer: B**

**NEW QUESTION 296**

Given:

```
interface Downloadable {
    public void download();
}

interface Readable extends Downloadable {           // line n1
    public void readBook();
}

abstract class Book implements Readable {           // line n2
    public void readBook() {
        System.out.println("Read Book");
    }
}

class EBook extends Book {                         // line n3
    public void readBook() {
        System.out.println("Read E-Book");
    }
}
```

And given the code fragment:

```
Book book1 = new EBook();
book1.readBook();
```

What is the result?

- A. Compilation fails at line n2
- B. Read Book
- C. Read E-Book
- D. Compilation fails at line n1
- E. Compilation fails at line n3

**Answer: B**

**NEW QUESTION 297**

Given the following class:

```
public class Rectangle {  
    private double length;  
    private double height;  
    private double area;  
  
    public void setLength(double length) {  
        this.length = length;  
    }  
    public void setHeight(double height) {  
        this.height = height;  
    }  
    public void setArea() {  
        area = length*height;  
    }  
}
```

Which two changes would encapsulate this class and ensure that the area field is always equal to length \* height whenever the Rectangle class is used? (Choose two.)

- A. Call the setArea method at the end of the setHeight method.
- B. Call the setArea method at the beginning of the setHeight method.
- C. Call the setArea method at the end of the setLength method.
- D. Call the setArea method at the beginning of the setLength method.
- E. Change the setArea method to private.
- F. Change the area field to public.

**Answer: AE**

#### **NEW QUESTION 298**

Given the code fragment:

```
13. List colors = new ArrayList();  
14. colors.add("green");  
15. colors.add("red");  
16. colors.add("blue");  
17. colors.add("yellow");  
18. colors.remove(2);  
19. colors.add(3, "cyan");  
20. System.out.print(colors);
```

What is the result?

- A. (green, red, yellow, cyan)
- B. (green, blue, yellow, cyan)
- C. (green, red, cyan, yellow)
- D. AnIndexOutOfBoundsException is thrown at runtime

**Answer: C**

**NEW QUESTION 299**

Given the code fragment:

```
abstract class Toy {  
    int price;  
    // line n1  
}
```

Which three code fragments are valid at line n1? (Choose three.)

- A. public static void insertToy() /\* code goes here \*/
- B. public abstract Toy getToy() {return new Toy();}
- C. public void printToy();
- D. public int calculatePrice() {return price;}
- E. public abstract int computeDiscount();

**Answer: CDE**

**NEW QUESTION 300**

Given:

```
public class Test {  
    int x, y;  
  
    public Test(int x, int y) {  
        initialize(x, y);  
    }  
  
    public void initialize(int x, int y) {  
        this.x = x * x;  
        this.y = y * y;  
    }  
  
    public static void main(String[] args) {  
        int x = 3, y = 5;  
        Test obj = new Test(x, y);  
        System.out.println(x + " " + y);  
    }  
}
```

What is the result?

- A. Compilation fails
- B. 3 5
- C. 0 0
- D. 9 25

**Answer: B**

**NEW QUESTION 301**

Given the code fragment:

```
public static void main(String[] args) {  
    int array[] = {10, 20, 30, 40, 50};  
    int x = array.length;  
    /* line n1 */  
}
```

Which two code fragments can be independently inserted at line n1 to enable the code to print the elements of the array in reverse order? (Choose two.)

- A. while (x > 0) {x--;System.out.print(array[x]);}
- B. do {x--;System.out.print(array[x]);} while (x >= 0);
- C. while (x >= 0) {System.out.print(array[x]);x--;}
- D. do {System.out.print(array[x]);--x;} while (x >= 0);
- E. while (x > 0) {System.out.print(array[--x]);}

**Answer: BE**

**NEW QUESTION 302**

.....

[Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps](#)

# Oracle

## Exam Questions 1Z0-808

Java SE 8 Programmer I



**NEW QUESTION 1**

Given:

Base.java:

```
class Base {  
    public void test(){  
        System.out.println("Base ");  
    }  
}
```

DerivedA.java:

```
class DerivedA extends Base {  
    public void test(){  
        System.out.println("DerivedA ");  
    }  
}
```

DerivedB.java:

```
class DerivedB extends DerivedA {  
    public void test(){  
        System.out.println("DerivedB ");  
    }  
    public static void main(String[] args) {  
        Base b1 = new DerivedB();  
        Base b2 = new DerivedA();  
        Base b3 = new DerivedB();  
        b1 = (Base) b3;  
        Base b4 = (DerivedA) b3;  
        b1.test();  
        b4.test();  
    }  
}
```

What is the result?

- A. BaseDerivedA
- B. BaseDerivedB
- C. DerivedBDerivedB
- D. DerivedBDerivedA
- E. A classcast Exception is thrown at runtime.

**Answer: C****NEW QUESTION 2**

Given the code fragment:

```
3. public static void main(String[] args) {  
4.     int iVar = 100;  
5.     float fVar = 100.100f;  
6.     double dVar = 123;  
7.     iVar = fVar;  
8.     fVar = iVar;  
9.     dVar = fVar;  
10.    fVar = dVar;  
11.    dVar = iVar;  
12.    iVar = dVar;  
13. }
```

Which three lines fail to compile?

- A. Line 7
- B. Line 8
- C. Line 9
- D. Line 10
- E. Line 11
- F. Line 12

**Answer:** ADF

#### NEW QUESTION 3

Given the code fragment:

```
public class Person {  
    String name;  
    int age = 25;  
  
    public Person(String name) {  
        this(); //line n1  
        setName(name);  
    }  
  
    public Person(String name, int age) {  
        Person(name); //line n2  
        setAge(age);  
    }  
    //setter and getter methods go here  
  
    public String show() {  
        return name + " " + age + " " + number ;  
    }  
    public static void main(String[] args) {  
        Person p1 = new Person("Jesse");  
        Person p2 = new Person("Walter", 52);  
        System.out.println(p1.show());  
        System.out.println(p2.show());  
    }  
}
```

What is the result?

- A. Jesse 25Walter 52
- B. Compilation fails only at line n1
- C. Compilation fails only at line n2

D. Compilation fails at both line n1 and line n2

**Answer:** D

**NEW QUESTION 4**

You are asked to create a method that accepts an array of integers and returns the highest value from that array.

Given the code fragment:

```
class Test {  
    public static void main (String [] args) {  
        int numbers [] = {12, 13, 42, 32, 15, 156, 23, 51, 12};  
        int max = findMax (numbers);  
    }  
/*line n1 */ {  
    int max = 0;  
    /* code goes here*/  
    return max;  
}  
}
```

Which method signature do you use at line n1?

- A. public int findMax (int [] numbers)
- B. static int[] findMax (int max)
- C. static int findMax (int [] numbers)
- D. final int findMax (int [] )

**Answer:** A

**NEW QUESTION 5**

Given:

```
public class SumTest {  
  
    public static void doSum(Integer x, Integer y) {  
        System.out.println("Integer sum is " + (x + y));  
    }  
  
    public static void doSum(double x, double y) {  
        System.out.println("double sum is " + (x + y));  
    }  
  
    public static void doSum(float x, float y) {  
        System.out.println("float sum is " + (x + y));  
    }  
  
    public static void doSum(int x, int y) {  
        System.out.println("int sum is " + (x + y));  
    }  
  
    public static void main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

What is the result?

- A) int sum is 30  
float sum is 30.0
- B) int sum is 30  
double sum is 30
- C) Integer sum is 30  
double sum is 30.0
- D) Integer sum is 30  
float sum is 30.0

- A. Option A  
B. Option B  
C. Option C  
D. Option D

Answer: B

#### NEW QUESTION 6

Given:

```
public class App {  
  
    public static void main(String[] args) {  
        Boolean[] bool = new Boolean[2];  
  
        bool[0] = new Boolean(Boolean.parseBoolean("true"));  
        bool[1] = new Boolean(null);  
  
        System.out.println(bool[0] + " " + bool[1]);  
    }  
}
```

What is the result?

- A. True false
- B. True null
- C. Compilation fails
- D. A NullPointerException is thrown at runtime

**Answer:** A

#### NEW QUESTION 7

Given the code fragment:

```
LocalDate Time dt= LocalDateTime.of (2014, 7, 31, 1, 1);  
dt.plusDays (30);  
dt. plusMonths (1);  
System.out.print (dt format (DateTimeFormatter. ISO_DATE) );
```

What is the result?

- A. An exception is thrown at runtime
- B. 07-31-2014
- C. 2014-07-31
- D. 2014-09-30

**Answer:** D

#### NEW QUESTION 8

Given the code fragment:

```
1. public class Test {  
2.     public static void main(String[] args) {  
3.         /* insert code here */  
4.         array[0]=10;  
5.         array[1]=20;  
6.         System.out.print(array[0]+": "+array[1]);  
7.     }  
8. }
```

Which code fragment, when inserted at line 3, enables the code to print 10:20?

- A. int[] array n= new int[2];
- B. int[] array;array = int[2];
- C. int array = new int[2];
- D. int array [2] ;

**Answer:** C

#### NEW QUESTION 9

Given the code fragment:

```
13. List colors = new ArrayList();
14. colors.add("green");
15. colors.add("red");
16. colors.add("blue");
17. colors.add("yellow");
18. colors.remove(2);
19. colors.add(3, "cyan");
20. System.out.print(colors);
```

What is the result?

- A. (green, red, yellow, cyan)
- B. (green, blue, yellow, cyan)
- C. (green, red, cyan, yellow)
- D. AnIndexOutOfBoundsException is thrown at runtime.

**Answer:** C

#### NEW QUESTION 10

Given the code fragment:

```
LocalDate date1 = LocalDate.now();
LocalDate date2 = LocalDate.of(2014, 6, 20);
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);
System.out.println("date1 = " + date1);
System.out.println("date2 = " + date2);
System.out.println("date3 = " + date3);
```

Assume that the system date is June 20, 2014. What is the result?

- A. Compilation fails.
- B. A DateParseException is thrown at runtime
- C. Date1 = 2014-05-20 Date2 = 2014-05-20 Date3 = 2014-05-20
- D. date1 = 06/20/2014 date2 = 2014-06-20 date3 = Jun 20, 2014

**Answer:** C

#### NEW QUESTION 10

Given the code fragment:

```
public static void main(String[] args) {
    StringBuilder sb = new StringBuilder(5);
    String s = "";

    if (sb.equals(s)) {
        System.out.println("Match 1");
    } else if (sb.toString().equals(s.toString())) {
        System.out.println("Match 2");
    } else {
        System.out.println("No Match");
    }
}
```

What is the result?

- A. Match 1
- B. Match 2
- C. No Match
- D. A NullPointerException is thrown at runtime.

**Answer:** B

#### NEW QUESTION 11

Given the code fragment:

```
Public static void main (String [] args) {  
    System.out.println ("Result A " + 0 + 1);  
    System.out.println ("Result B " + (1) + (2) );  
}
```

What is the result?

- A. Result A 1  
    Result B 3
- B. Result A 01  
    Result B 3
- C. Result A 01  
    Result B 12
- D. Result A 1  
    Result B 12

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** C

#### **NEW QUESTION 12**

Given the following array:

```
int[] intArr = { 8, 16, 32, 64, 128 };
```

Which two code fragments, independently, print each element in this array?

- A) 

```
for (int i : intArr) {
    System.out.print(intArr[i] + " ");
}
```
- B) 

```
for (int i : intArr) {
    System.out.print(i + " ");
}
```
- C) 

```
for (int i=0 : intArr) {
    System.out.print(intArr[i] + " ");
    i++;
}
```
- D) 

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(i + " ");
}
```
- E) 

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(intArr[i] + " ");
}
```
- F) 

```
for (int i; i < intArr.length; i++) {
    System.out.print(intArr[i] + " ");
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

Answer: BE

#### NEW QUESTION 15

Given:

```
System.out.println("5 + 2 = " + 3 + 4);
System.out.println("5 + 2 = " + (3 + 4));
```

What is the result?

- A)  $5 + 2 = 34$   
 $5 + 2 = 34$
- B)  $5 + 2 + 3 + 4$   
 $5 + 2 = 7$
- C)  $7 = 7$   
 $7 + 7$
- D)  $5 + 2 = 34$   
 $5 + 2 = 7$

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** D

#### NEW QUESTION 20

Given the code fragment:

```
public static void main(String[] args) {  
    String str = " ";  
    str.trim();  
    System.out.println(str.equals("") + " " + str.isEmpty());  
}
```

What is the result?

- A. true true
- B. true false
- C. false false
- D. false true

**Answer:** C

#### NEW QUESTION 25

Given:

```
class Equal {  
    public static void main (String [] args) {  
        String str1 = "Java";  
        String [] str2 = { "J", "a", "v", "a" };  
        String str3 = "";  
        for (String str : str2) {  
            str3 = str3+str;  
        }  
        boolean b1 = (str1== str3);  
        boolean b2 = (str1.equals (str3));  
        System.out.print (b1+", "+b2);  
    }  
}
```

What is the result?

- A. false, false
- B. false, true
- C. true, false
- D. true, true

**Answer:** B

#### NEW QUESTION 27

Given:

```
public class Fieldinit {  
    char c;  
    boolean b;  
    float f;  
    void printAll() {  
        System.out.println ("c = " + c);  
        System.out.println ("b = " + b);  
        System.out.println ("f = " + f);  
    }  
    public static void main (String [] args) {  
        FieldInit f = new FieldInit ();  
        f.printAll ();  
    }  
}
```

What is the result?

- A. c=b = falsef = 0.0
- B. c= nullb = truef = 0.0
- C. c=0b = falsef = 0.0f
- D. c= nullb = falsef = 0.0F

**Answer:** C

#### NEW QUESTION 28

Given the code fragment:

```
public static void main (String[ ] args) {  
    int data [ ] = {2010, 2013, 2014, 2015, 2014};  
    int key = 2014;  
    int count = 0;  
    for (int e: data) {  
        if (e! = key) {  
            continue;  
            count++;  
        }  
    }  
    System.out.print (count + "Found");  
}
```

What is the result?

- A. Compilation fails.
- B. 0 Found
- C. 1 Found
- D. 3 Found

**Answer:** D

#### NEW QUESTION 32

Which three statements describe the object-oriented features of the Java language?

- A. Objects cannot be reused.
- B. A subclass can inherit from a superclass.
- C. Objects can share behaviors with other objects.
- D. A package must contain more than one class.
- E. Object is the root class of all other objects.
- F. A main method must be declared in every class.

**Answer:** BCF

#### NEW QUESTION 37

Given the code fragments:

**A.java:**

```
package p1;  
public class A {  
}
```

**B.java:**

```
package p1.p2;  
//line n1  
public class B {  
    public void doStuff () {  
        A b = new A ();  
    }  
}
```

**C.java**

```
package p3;  
//line n2  
public class C {  
    public static void main (String [] args) {  
        A 01 = new A ();  
        B 02 = new B ();  
    }  
}
```

Which modification enables the code to compile?

- A. Replace line n1 with:`import p1.*;`Replace line n2 with:`import p1. p2.*;`
- B. Replace line n1 with:`import p1. A;`Replace line n2 with:`import p1.*;`
- C. Replace line n1 with:`import p1. A;`Replace line n2 with:`import p1. A;import p1. p2.B ;`
- D. Replace line n1 with:`import p1;`Replace line n2 with:`import p1;import p1. p2;`

**Answer:** C

#### NEW QUESTION 38

Which statement is true about Java byte code?

- A. It can run on any platform.
- B. It can run on any platform only if it was compiled for that platform.
- C. It can run on any platform that has the Java Runtime Environment.
- D. It can run on any platform that has a Java compiler.

E. It can run on any platform only if that platform has both the Java Runtime Environment and a Java compiler.

**Answer:** ACDE

**NEW QUESTION 42**

Given the code fragment:

```
public static void main (String [] args) {  
    String names [] = {"Thomas", "Peter", "Joseph");  
    String pws [] = new String [3];  
    int idx = 0;  
    try {  
        for (String n: names) {  
            pws [idx] = n.substring (2, 6);  
            idx++;  
        }  
    }  
    catch (Exception e) {  
        System.out.println ("Invalid Name");  
    }  
    for (String p: pws) {  
        System.out.println (p);  
    }  
}
```

What is the result?

- A. Invalid Name
- B. Invalid Nameomas
- C. Invalid Name omas null null
- D. omasterseph

**Answer:** C

**NEW QUESTION 46**

Given the following class:

```
public class Rectangle {  
    private double length;  
    private double height;  
    private double area;  
  
    public void setLength(double length) {  
        this.length = length;  
    }  
    public void setHeight(double height) {  
        this.height = height;  
    }  
    public void setArea() {  
        area = length*height;  
    }  
}
```

Which two changes would encapsulate this class and ensure that the area field is always equal to length \* height whenever the Rectangle class is used?

- A. Call the setArea method at the end of the setHeight method.
- B. Call the setArea method at the beginning of the setHeight method.
- C. Call the setArea method at the end of the setLength method.
- D. Call the setArea method at the beginning of the setLength method.
- E. Change the setArea method to private.
- F. Change the area field to public.

**Answer:** AE

#### NEW QUESTION 47

Given the following classes:

```
public class Employee {  
    public int salary;  
}  
  
public class Manager extends Employee {  
    public int budget;  
}  
  
public class Director extends Manager {  
    public int stockOptions;  
}
```

And given the following main method:

```
public static void main(String[] args) {  
    Employee employee = new Employee();  
    Manager manager = new Manager();  
    Director director = new Director();  
    //line n1  
}
```

Which two options fail to compile when placed at line n1 of the main method?

- A. employee.salary = 50\_000;
- B. director.salary = 80\_000;
- C. employee.budget = 200\_000;
- D. manager.budget = 1\_000\_000;

E. manager.stockOption = 500;  
F. director.stockOptions = 1\_000;

Answer: CE

**NEW QUESTION 49**

You are developing a banking module. You have developed a class named ccMask that has a maskCC method. Given the code fragment:

```
class CCmask {  
    public static String maskCC(String creditCard) {  
        String x = "XXXX-XXXX-XXXX-";  
        //line n1  
    }  
  
    public static void main(String[] args) {  
        System.out.println(maskCC("1234-5678-9101-1121"));  
    }  
}
```

You must ensure that the maskCC method returns a string that hides all digits of the credit card number except the four last digits (and the hyphens that separate each group of four digits).

Which two code fragments should you use at line n1, independently, to achieve this requirement?

- A) `StringBuilder sb = new StringBuilder(creditCard);  
sb.substring(15, 19);  
return x + sb;`
- B) `return x + creditCard.substring(15, 19);`
- C) `StringBuilder sb = new StringBuilder(x);  
sb.append(creditCard, 15, 19);  
return sb.toString();`
- D) `StringBuilder sb = new StringBuilder(creditCard);  
StringBuilder s = sb.insert(0, x);  
return s.toString();`

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: BC

**NEW QUESTION 54**

Given:

```
interface Readable {  
    public void readBook();  
    public void setBookMark();  
}  
  
abstract class Book implements Readable { // line n1  
    public void readBook() {}  
    // line n2  
}  
  
class EBook extends Book { // line n3  
    public void readBook() {}  
    // line n4  
}
```

And given the code fragment: Book book1 = new EBook (); Book1.readBook();  
Which option enables the code to compile?

- A. Replace the code fragment at line n3 with:  
abstract class EBook extends Book {
- B. Replace the code fragment at line n1 with:  
class Book implements Readable {
- C. At line n2 insert:  
public abstract void setBookMark ();
- D. At line n4 insert:  
public void setBookMark () {}

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

**NEW QUESTION 57**

Given:

```
public class App {  
  
    String myStr = "7007";  
  
    public void doStuff(String str) {  
        int myNum = 0;  
        try {  
            String myStr = str;  
            myNum = Integer.parseInt(myStr);  
        } catch (NumberFormatException ne) {  
            System.err.println("Error");  
        }  
        System.out.println(  
            "myStr: " + myStr + ", myNum: " + myNum);  
    }  
  
    public static void main(String[] args) {  
        App obj = new App();  
        obj.doStuff("9009");  
    }  
}
```

What is the result?

- A. myStr: 9009, myNum: 9009
- B. myStr: 7007, myNum: 7007
- C. myStr: 7007, myNum: 9009
- D. Compilation fails

**Answer:** C

#### NEW QUESTION 61

Given:

```
public class Test {  
  
    public static void main(String[] args) {  
        if (args[0].equals("Hello") ? false : true) {  
            System.out.println("Success");  
        } else {  
            System.out.println("Failure");  
        }  
    }  
}
```

And given the commands: javac Test.java  
Java Test Hello What is the result?

- A. Success
- B. Failure
- C. Compilation fails.
- D. An exception is thrown at runtime

**Answer:** B

#### NEW QUESTION 63

Given the code fragment:

```
String shirts[][] = new String[2][2];
shirts[0][0] = "red";
shirts[0][1] = "blue";
shirts[1][0] = "small";
shirts[1][1] = "medium";
```

Which code fragment prints red: blue: small: medium?

- C A) 

```
for (int index = 1; index < 2; index++) {
    for (int idx = 1; idx < 2; idx++) {
        System.out.print(shirts[index][idx] + ":");
    }
}
```
- C B) 

```
for (int index = 0; index < 2; ++index) {
    for (int idx = 0; idx < index; ++idx) {
        System.out.print(shirts[index][idx] + ":");
    }
}
```
- C C) 

```
for (String c : colors) {
    for (String s : sizes) {
        System.out.println(s + ":");
    }
}
```
- C D) 

```
for (int index = 0; index < 2;) {
    for (int idx = 0; idx < 2;) {
        System.out.print(shirts[index][idx] + ":");
        idx++;
    }
    index++;
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** D

**NEW QUESTION 67**

Given the code fragment:

```
abstract class Planet {  
    protected void revolve() { //line n1  
    }  
  
    abstract void rotate(); //line n2  
}  
  
class Earth extends Planet {  
    void revolve() { //line n3  
    }  
  
    protected void rotate() { //line n4  
    }  
}
```

Which two modifications, made independently, enable the code to compile?

- A. Make the method at line n1 public.
- B. Make the method at line n2 public.
- C. Make the method at line n3 public.
- D. Make the method at line n3 protected.
- E. Make the method at line n4 public.

**Answer:** BC

#### NEW QUESTION 71

Given:

```
public class MyClass {  
    public static void main(String[] args) {  
        String s = "Java Duke";  
        int len = s.trim().length();  
        System.out.print(len);  
    }  
}
```

What is the result?

- A. Compilation fails.
- B. 11
- C. 8
- D. 9
- E. 10

**Answer:** D

#### NEW QUESTION 76

Given the following code:

```
int[] intArr = {15, 30, 45, 60, 75};  
intArr[2] = intArr[4];  
intArr[4] = 90;
```

What are the values of each element in intArr after this code has executed?

- A. 15, 60, 45, 90, 75
- B. 15, 90, 45, 90, 75
- C. 15, 30, 75, 60, 90
- D. 15, 30, 90, 60, 90
- E. 15, 4, 45, 60, 90

**Answer:** C

**NEW QUESTION 79**

Given the code fragment:

```
public static void main (String [ ] args) {  
    int [] stack = {10,20,30}  
    int size = 3;  
    int i dx = 0;  
    /*line n1 */  
    System.out.print ("The Top element: " + stack [idx] );  
}
```

Which code fragment, inserted at line n1, prints The Top element: 30?

- A. do {  
 idx++;  
 } while (idx >= size);
- B. while (idx < size) {  
 idx++;  
 }
- C. do {  
 idx++;  
 } while (idx < size -1);
- D. do {  
 idx++;  
 } while (idx <= size);
- E. while (idx <= size -1) {  
 idx++  
 }

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** A**NEW QUESTION 81**

Given the code fragment:

```
if (aVar++ < 10) {  
    System.out.println(aVar + " Hello World!");  
} else {  
    System.out.println(aVar + " Hello Universe!");  
}
```

What is the result if the integer aVar is 9?

- A. Compilation fails.
- B. 10 Hello Universe!

- C. 10 Hello World!  
D. 9 Hello World!

Answer: C

#### NEW QUESTION 86

You are asked to develop a program for a shopping application, and you are given the following information: Which definition of the Toy class adds a valid layer of abstraction to the class hierarchy?

- A) 

```
public abstract class Toy{  
    public abstract int calculatePrice(Toy t);  
    public void printToy(Toy t) { /* code goes here */ }  
}
```
- B) 

```
public abstract class Toy {  
    public int calculatePrice(Toy t) ;  
    public void printToy(Toy t) ;  
}
```
- C) 

```
public abstract class Toy {  
    public int calculatePrice(Toy t);  
    public final void printToy(Toy t){ /* code goes here */ }  
}
```
- D) 

```
public abstract class Toy {  
    public abstract int calculatePrice(Toy t) { /* code goes here */ }  
    public abstract void printToy(Toy t) { /* code goes here */ }  
}
```

- A. Option A  
B. Option B  
C. Option C  
D. Option D

Answer: A

#### NEW QUESTION 88

Given: Acc.java:

```
package p1;  
public class Acc {  
    int p;  
    private int q;  
    protected int r;  
    public int s;  
}
```

Test.java:

```
package p2;  
import p1.Acc;  
public class Test extends Acc {  
    public static void main(String[] args) {  
        Acc obj = new Test();  
    }  
}
```

Which statement is true?

- A. Both p and s are accessible by obj.  
B. Only s is accessible by obj.

- C. Both r and s are accessible by obj.
- D. p, r, and s are accessible by obj.

Answer: B

#### NEW QUESTION 90

Given:

```
public class Test {  
    public static void main(String[] args) {  
        Test ts = new Test();  
        System.out.print(isAvailable + " ");  
        isAvailable= ts.doStuff();  
        System.out.println(isAvailable);  
    }  
    public static boolean doStuff() {  
        return !isAvailable;  
    }  
    static boolean isAvailable = false;  
}
```

What is the result?

- A. Compilation fails.
- B. false true
- C. true false
- D. true true
- E. false false

Answer: B

#### NEW QUESTION 94

Which two statements are true?

- A. Error class is unextendable.
- B. Error class is extendable.
- C. Error is a RuntimeException.
- D. Error is an Exception.
- E. Error is a Throwable.

Answer: BC

#### NEW QUESTION 97

Given:

```
public class MarkList {  
    int num;  
    public static void graceMarks(MarkList obj4) {  
        obj4.num += 10;  
    }  
    public static void main(String[] args) {  
        MarkList obj1 = new MarkList();  
        MarkList obj2 = obj1;  
        MarkList obj3 = null;  
        obj2.num = 60;  
        graceMarks(obj2);  
    }  
}
```

How many MarkList instances are created in memory at runtime?

- A. 1
- B. 2

- C. 3  
D. 4

Answer: A

**NEW QUESTION 102**

Given:

```
public class MyField {  
    int x;  
    int y;  
    public void doStuff(int x, int y) {  
        this.x = x;  
        y = this.y;  
    }  
    public void display () {  
        System.out.print(x + " " + y + " : ");  
    }  
    public static void main(String[] args) {  
        MyField m1 = new MyField();  
        m1.x = 100;  
        m1.y = 200;  
        MyField m2 = new MyField();  
        m2.doStuff(m1.x, m1.y);  
        m1.display();  
        m2.display();  
    }  
}
```

What is the result?

- A. 100 0 : 100 200:  
B. 100 0 : 100 0 :  
C. 100 200 : 100 200 :  
D. 100 200 : 100 0 :

Answer: B

**NEW QUESTION 106**

Given:

```
public class Test {  
    int x, y;  
  
    public Test(int x, int y) {  
        initialize(x, y);  
    }  
  
    public void initialize(int x, int y) {  
        this.x = x * x;  
        this.y = y * y;  
    }  
  
    public static void main(String[] args) {  
        int x = 3, y = 5;  
        Test obj = new Test(x, y);  
        System.out.println(x + " " + y);  
    }  
}
```

What is the result?

- A. Compilation fails.
- B. 3 5
- C. 0 0
- D. 9 25

**Answer:** B

#### NEW QUESTION 110

Given:

```
class Vehicle {  
    int x;  
    Vehicle() {  
        this(10); // line n1  
    }  
    Vehicle(int x) {  
        this.x = x;  
    }  
}  
  
class Car extends Vehicle {  
    int y;  
    Car() {  
        super();  
        this(20); // line n2  
    }  
    Car(int y) {  
        this.y = y;  
    }  
    public String toString() {  
        return super.x + ":" + this.y;  
    }  
}
```

And given the code fragment:

And given the code fragment:

```
Vehicle y = new Car();  
System.out.println(y);
```

What is the result?

- A. 10:20
- B. 0:20
- C. Compilation fails at line n1
- D. Compilation fails at line n2

**Answer: D**

#### **NEW QUESTION 111**

Given the code fragment:

```
public class Test {  
    public static void main(String[] args) {  
        //line n1  
        switch (x) {  
            case 1:  
                System.out.println("One");  
                break;  
            case 2:  
                System.out.println("Two");  
                break;  
        }  
    }  
}
```

Which three code fragments can be independently inserted at line n1 to enable the code to print one?

- A. Byte x = 1;
- B. short x = 1;
- C. String x = "1";
- D. Long x = 1;
- E. Double x = 1;
- F. Integer x = new Integer ("1");

**Answer:** ABF

#### NEW QUESTION 114

Given:

```
public class App {  
    int count;  
    public static void displayMsg () {  
        count++;  
        // line n1  
        System.out.println ("Welcome +" + "Visit Count: "+count); // line n2  
    }  
    public static void main (String [] args) {  
        App.displayMsg ();  
        // line n3  
        App.displayMsg ();  
        // line n4  
    }  
}
```

What is the result?

- A. Compilation fails at line n3 and line n4.
- B. Compilation fails at line n1 and line n2.
- C. Welcome Visit Count:1Welcome Visit Count: 2
- D. Welcome Visit Count:1Welcome Visit Count: 2

**Answer:** B

#### NEW QUESTION 118

Given the code fragment from three files:

SalesMan.java:

```
package sales;
public class SalesMan { }
```

Product.java:

```
package sales.products;
public class Product { }
```

Market.java:

```
1. package market;
2. // insert code here
3. public class USMarket {
4.     SalesMan sm;
5.     Product p;
6. }
```

Which code fragment, when inserted at line 2, enables the code to compile?

- A) import sales.\*;
- B) import java.sales.products.\*;
- C) import sales;  
 import sales.products;
- D) import sales.\*;  
 import products.\*;
- E) import sales.\*;  
 import sales.products.\*;

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

Answer: E

#### NEW QUESTION 119

Given the code fragment: int[] array = {1, 2, 3, 4, 5}; And given the requirements:

1. Process all the elements of the array in the order of entry.
2. Process all the elements of the array in the reverse order of entry.
3. Process alternating elements of the array in the order of entry. Which two statements are true?

- A. Requirements 1, 2, and 3 can be implemented by using the enhanced for loop.
- B. Requirements 1, 2, and 3 can be implemented by using the standard for loop.
- C. Requirements 2 and 3 CANNOT be implemented by using the standard for loop.
- D. Requirement 1 can be implemented by using the enhanced for loop.
- E. Requirement 3 CANNOT be implemented by using either the enhanced for loop or the standard for loop.

Answer: BD

#### NEW QUESTION 122

Given:

```
class X {  
    static int i;  
    int j;  
    public static void main(String[] args) {  
        X x1 = new X();  
        X x2 = new X();  
        x1.i = 3;  
        x1.j = 4;  
        x2.i = 5;  
        x2.j = 6;  
        System.out.println(  
            x1.i + " " +  
            x1.j + " " +  
            x2.i + " " +  
            x2.j);  
    }  
}
```

What is the result?

- A. 3 4 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 4 6

Answer: C

#### NEW QUESTION 123

Given:

```
public class Vowel {  
    private char var;  
    public static void main(String[] args) {  
        char var1 = 'a';  
        char var2 = var1;  
        var2 = 'e';  
  
        Vowel obj1 = new Vowel ();  
        Vowel obj2 = obj1;  
        obj1.var = 'i';  
        obj2.var = 'o';  
  
        System.out.println(var1 + ", " +var2);  
        System.out.print(obj1.var + ", " +obj2.var);  
    }  
}
```

What is the result?

- A. a, oi, o
- B. a, oo, o
- C. o, oi, o
- D. o, oo, o

Answer: B

#### NEW QUESTION 125

Given the code fragment:

```
public static void main(String[] args) {  
    String[] arr = {"A", "B", "C", "D"};  
    for (int i = 0; i < arr.length; i++) {  
        System.out.print(arr[i] + " ");  
        if (arr[i].equals("C")) {  
            continue;  
        }  
        System.out.println("Work done");  
        break;  
    }  
}
```

What is the result?

- A. A B C Work done
- B. A B C D Work done
- C. A Work done
- D. Compilation fails

**Answer:** C

#### NEW QUESTION 126

Given the code fragment:

```
public class Employee {  
    String name;  
    boolean contract;  
    double salary;  
    Employee() {  
        // line n1  
    }  
    public String toString() {  
        return name + ":" + contract + ":" + salary;  
    }  
    public static void main(String[] args) {  
        Employee e = new Employee();  
        // line n2  
        System.out.print(e);  
    }  
}
```

Which two modifications, when made independently, enable the code to print joe:true: 100.0?

A) Replace line n2 with:

```
e.name = "Joe";  
e.contract = true;  
e.salary = 100;
```

B) Replace line n2 with:

```
this.name = "Joe";  
this.contract = true;  
this.salary = 100;
```

C) Replace line n1 with:

```
this.name = new String("Joe");  
this.contract = new Boolean(true);  
this.salary = new Double(100);
```

D) Replace line n1 with:

```
name = "Joe";  
contract = TRUE;  
salary = 100.0f;
```

E) Replace line n1 with:

```
this("Joe", true, 100);
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** AC

#### NEW QUESTION 128

Given:

```
interface Downloadable {  
    public void download();  
}  
  
interface Readable extends Downloadable { // line n1  
    public void readBook();  
}  
  
abstract class Book implements Readable { // line n2  
    public void readBook() {  
        System.out.println("Read Book");  
    }  
}  
  
class EBook extends Book { // line n3  
    public void readBook() {  
        System.out.println("Read E-Book");  
    }  
}
```

And given the code fragment:

```
Book book1 = new EBook();  
book1.readBook();
```

What is the result?

- A. Compilation fails at line n2.
- B. Read Book
- C. Read E-Book
- D. Compilation fails at line n1.
- E. Compilation fails at line n3.

**Answer:** B

#### NEW QUESTION 133

Given:

```
interface Readable {  
    public void readBook();  
    public void setBookMark();  
}  
  
abstract class Book implements Readable { // line n1  
    public void readBook() {}  
    // line n2  
}  
  
class EBook extends Book { // line n3  
    public void readBook() {}  
    // line n4  
}
```

Which option enables the code to compile?

C) A) Replace the code fragment at line n1 with:

```
class Book implements Readable {
```

C) B) At line n2 insert:

```
public abstract void setBookMark();
```

C) C) Replace the code fragment at line n3 with:

```
abstract class EBook extends Book {
```

C) D) At line n4 insert:

```
public void setBookMark() { }
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** C

#### NEW QUESTION 135

Given:

```
public class App {  
    public static void main(String[] args) {  
        int i = 10;  
        int j = 20;  
        int k = j += i / 5;  
        System.out.print(i + " : " + j + " : " + k);  
    }  
}
```

What is the result?

- A. 10 : 30 : 6
- B. 10 : 22 : 22
- C. 10 : 22 : 20
- D. 10 : 22 : 6

**Answer:** B

**Explanation:**

Your Code ...

```
1 public class App {  
2     public static void main (String[] args) {  
3         int i = 10;  
4         int j = 20;  
5         int k = j += i / 5;  
6         System.out.print (i + " : " + j + " : " + k);  
7     }  
8 }  
9
```

External Libraries ... 

CommandLine Arguments ...

Interactive mode :  OFF

Version:

JDK 9.0.1

Stdin Inputs...

Result...

CPU Time: 0.20 sec(s), Memory: 32080 kilobyte(s)

compiled and executed in 1.229 sec(s)

10 : 22 : 22

**NEW QUESTION 139**

Given the code fragments:

Person.java:

```
public class Person {  
    String name;  
    int age;  
  
    public Person(String n, int a) {  
        name = n;  
        age = a;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getAge() {  
        return age;  
    }  
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {  
    for (Person p : list) {  
        if (predicate.test(p)) {  
            System.out.println(p.name + " ");  
        }  
    }  
}  
  
public static void main(String[] args) {  
    List<Person> iList = Arrays.asList(new Person("Hank", 45),  
   new Person("Charlie", 40),  
   new Person("Smith", 38));  
    //line n1  
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

- A. checkAge (iList, () ->
- B. get Age () > 40);
- C. checkAge(iList, Person p -> p.getAge() > 40);
- D. checkAge (iList, p -> p.getAge () > 40);
- E. checkAge(iList, (Person p) -> { p.getAge() > 40; });

**Answer: C**

#### NEW QUESTION 141

fragment:

```
1. class X {  
2.     public void printFileContent() {  
3.         /* code goes here */  
4.         throw new IOException();  
5.     }  
6. }  
7. public class Test {  
8.     public static void main(String[] args) {  
9.         X xobj = new X();  
10.        xobj.printFileContent();  
11.    }  
12. }
```

Which two modifications should you make so that the code compiles successfully?

- A) Replace line 8 with `public static void main(String[] args) throws Exception {`
- B) Replace line 10 with:  

```
try {  
    xobj.printFileContent();  
}  
catch(Exception e) {}  
catch(IOException e) {}
```
- C) Replace line 2 with `public void printFileContent() throws IOException {`
- D) Replace line 4 with `throw IOException("Exception raised");`
- E) At line 11, insert `throw new IOException();`

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** AC

**NEW QUESTION 146**

Given:

```
class Vehicle {  
    String type = "4W";  
    int maxSpeed = 100;  
  
    Vehicle(String type, int maxSpeed) {  
        this.type = type;  
        this.maxSpeed = maxSpeed;  
    }  
}  
  
class Car extends Vehicle {  
    String trans;  
  
    Car(String trans) { //line n1  
        this.trans = trans;  
    }  
  
    Car(String type, int maxSpeed, String trans) {  
        super(type, maxSpeed);  
        this(trans); //line n2  
    }  
}
```

And given the code fragment:

7. Car c1 = new Car("Auto");
8. Car c2 = new Car("4W", 150, "Manual");
9. System.out.println(c1.type + " " + c1.maxSpeed + " " + c1.trans);
10. System.out.println(c2.type + " " + c2.maxSpeed + " " + c2.trans);

What is the result?

- A. 4W 100 Auto4W 150 Manual
- B. Null 0 Auto4W 150 Manual
- C. Compilation fails only at line n1
- D. Compilation fails only at line n2
- E. Compilation fails at both line n1 and line n2

**Answer: C**

#### NEW QUESTION 150

Given:

```
class Animal {  
    String type = "Canine";  
    int maxSpeed = 60;  
  
    Animal () {}  
  
    Animal (String type, int maxSpeed) {  
        this.type = type;  
        this.maxSpeed = maxSpeed;  
    }  
}  
  
class WildAnimal extends Animal {  
    String bounds;  
  
    WildAnimal (String bounds) {  
        //line n1  
    }  
    WildAnimal (String type, int maxSpeed,  
        //line n2  
    }  
}
```

And given the code fragment:

7. WildAnimal wolf = new WildAnimal ("Long");
8. WildAnimal tiger = new WildAnimal ("Feline", 80, "Short");
9. System.out.println (wolf.type + " " + wolf.maxSpeed + " " + wolf.bounds);
10. System.out.println (tiger.type + " " + tiger.maxSpeed + " " + tiger.bounds);

Which two modifications enable the code to print the following output? Canine 60 Long  
Feline 80 Short

- A. Replace line n1 with:super ();this.bounds = bounds;
- B. Replace line n1 with:this.bounds = bounds;super ();
- C. Replace line n2 with:super (type, maxSpeed);this (bounds);
- D. Replace line n1 with:this ("Canine", 60);this.bounds = bounds;
- E. Replace line n2 with:super (type, maxSpeed);this.bounds = bounds;

**Answer:** A

#### NEW QUESTION 153

The following grid shows the state of a 2D array:

|   |   |   |
|---|---|---|
| 0 | 0 |   |
| X |   | 0 |
| X |   | X |

This grid is created with the following code:

```
char[][] grid = new char[3][3];
grid[1][1] = 'X';
grid[0][0] = 'O';
grid[2][1] = 'X';
grid[0][1] = 'O';
grid[2][2] = 'X';
grid[1][2] = 'O';
```

Which line of code, when inserted in place of //line n1, adds an X into the grid so that the grid contains three consecutive X's?

- A. grid[1][3] = 'X';
- B. grid[3][1] = 'X';
- C. grid[0][2] = 'X';
- D. grid[2][0] = 'X';
- E. grid[1][2] = 'X';

Answer: C

#### NEW QUESTION 158

Given the following main method:

```
public static void main(String[] args) {
    int num = 5;
    do {
        System.out.print(num-- + " ");
    } while (num == 0);
}
```

What is the result?

- A. 5 4 3 2 1 0
- B. 5 4 3 2 1
- C. 4 2 1
- D. 5
- E. Nothing is printed

Answer: D

#### NEW QUESTION 159

Given the following two classes:

```
public class Customer {
    ElectricAccount acct = new ElectricAccount();

    public void useElectricity(double kWh) {
        acct.addKWh(kWh);
    }
}

public class ElectricAccount {
    private double kWh;
    private double rate = 0.07;
    private double bill;

    //line n1
}
```

How should you write methods in the ElectricAccount class at line n1 so that the member variable bill is always equal to the value of the member variable kwh multiplied by the member variable rate?

Any amount of electricity used by a customer (represented by an instance of the customer class) must contribute to the customer's bill (represented by the member variable bill) through the method use Electricity method. An instance of the customer class should never be able to tamper with or decrease the value of the member variable bill.

C A) public void addKWh(double kWh) {  
    this.kWh += kWh;  
    this.bill = this.kWh\*this.rate;  
}  
  
C B) public void addKWh(double kWh) {  
    if (kWh > 0){  
        this.kWh += kWh;  
        this.bill = this.kWh \* this.rate;  
    }  
}  
  
C C) private void addKWh(double kWh) {  
    if (kWh > 0) {  
        this.kWh += kWh;  
        this.bill = this.kWh\*this.rate;  
    }  
}  
  
C D) public void addKWh(double kWh) {  
    if(kWh > 0) {  
        this.kWh += kWh;  
        setBill(this.kWh);  
    }  
}  
public void setBill(double kWh) {  
    bill = kWh\*rate;  
}

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: AC

#### NEW QUESTION 162

Given the code fragment:

```
24. float var1 = (12_345.01 >= 123_45.00) ? 12_456 : 124_56.02f;  
25. float var2 = var1 + 1024;  
26. System.out.print(var2);
```

What is the result?

- A. An exception is thrown at runtime.
- B. Compilation fail
- C. 13480.0
- D. 13480.02

Answer: C

#### NEW QUESTION 164

Given the following class:

```
public class CheckingAccount {  
    public int amount:  
        // line n1  
}
```

And given the following main method, located in another class:

```
public static void main (String [] args) {  
    CheckingAccount acct = new CheckingAccount ();  
    //line n2  
}
```

Which three pieces of code, when inserted independently, set the value of amount to 100?

A. At line n2 insert:

```
    amount = 100;
```

B. At line n2 insert:

```
    This. amount = 100
```

C. At line n2 insert:

```
    acct.amount = 100
```

D. At line n1 insert:

```
    public CheckingAccount () {  
        amount = 100;  
    }
```

E. At line n1 insert:

```
    public CheckingAccount () {  
        this.amount = 100;  
    }
```

F. At line n1 insert:

```
    public CheckingAccount () {  
        acct.amount = 100;  
    }
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

**Answer:** BCE

**NEW QUESTION 165**

Given:

```
public class TestScope {  
    public static void main(String[] args) {  
        int var1 = 200;  
        System.out.print(doCalc(var1));  
        System.out.print(" " + var1);  
    }  
    static int doCalc(int var1){  
        var1 = var1 * 2;  
        return var1;  
    }  
}
```

What is the result?

- A. 400 200
- B. 200 200
- C. 400 400
- D. Compilation fails.

**Answer:** A**NEW QUESTION 169**

Given:

```
class Student {  
    String name;  
    public Student(String name) {  
        this.name = name;  
    }  
}  
  
public class Test {  
    public static void main(String[] args) {  
        Student[] students = new Student[3];  
        students[1] = new Student("Richard");  
        students[2] = new Student("Donald");  
        for (Student s : students) {  
            System.out.println(" " + s.name);  
        }  
    }  
}
```

What is the result?

- A. nullRichardDonald
- B. RichardDonald
- C. Compilation fails.
- D. AnArrayIndexOutOfBoundsException is thrown at runtime.
- E. ANullPointerException is thrown at runtime.

**Answer:** A**NEW QUESTION 172**

Given:

```
class A {  
    public void test () {  
        System.out.println ("A");  
    }  
}  
class B extends A {  
    public void test () {  
        System.out.println ("B");  
    }  
}  
public class C extends A {  
    public void test () {  
        System.out.println ("C");  
    }  
  
    public static void main (String [] args) {  
        A b1 = new A ();  
        A b2 = new C ();  
        b1 = (A) b2;  
        A b3 = (B) b2; //line n1  
        A b3 = (B) b2; //line n2  
        b1.test ();  
        b3.test ();  
    }  
}
```

What is the result?

- A. AB
- B. AC
- C. CC
- D. A ClassCastException is thrown only at line n1.
- E. A ClassCastException is thrown only at line n2.

**Answer:** E

#### NEW QUESTION 175

Given:

```
public class Test {  
    public static void main(String[] args) {  
        boolean a = new Boolean(Boolean.valueOf (args[0]));  
        boolean b = new Boolean(args[1]);  
        System.out.println(a + " " + b);  
    }  
}
```

And given the commands: javac Test.java  
java Test TRUE null What is the result?

- A. TRUE null
- B. true false
- C. false false
- D. true true
- E. A ClassCastException is thrown at runtime.

**Answer:** D

#### NEW QUESTION 179

Given:

```
public class Test {  
  
    public static void main(String[] args) {  
  
        String[][] chs = new String[2][];  
        chs[0] = new String[2];  
        chs[1] = new String[5];  
        int i = 97;  
  
        for (int a = 0; a < chs.length; a++) {  
            for (int b = 0; b < chs.length; b++) {  
                chs[a][b] = "" + i;  
                i++;  
            }  
        }  
  
        for (String[] ca : chs) {  
            for (String c : ca) {  
                System.out.print(c + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

What is the result?

- A. 97 98 99 100 null null null
- B. 97 98 99 100 101 102 103
- C. Compilation fails.
- D. A NullPointerException is thrown at runtime.
- E. An ArrayIndexOutOfBoundsException is thrown at runtime.

**Answer:** A

#### NEW QUESTION 183

Which two are benefits of polymorphism?

- A. Faster code at runtime
- B. More efficient code at runtime
- C. More dynamic code at runtime
- D. More flexible and reusable code
- E. Code that is protected from extension by other classes

**Answer:** BC

#### NEW QUESTION 187

Which statement is true about the switch statement?

- A. It must contain the default section.
- B. The break statement, at the end of each case block, is mandatory.
- C. Its case label literals can be changed at runtime.

D. Its expression must evaluate to a single value.

**Answer:** D

**NEW QUESTION 192**

Given the code fragment:

```
public static void main (String [] args) {  
    ArrayList<Integer> points = new ArrayList<> ();  
    points.add (1);  
    points.add (2);  
    points.add (3);  
    points.add (4);  
    points.add (null);  
    points.remove (2);  
    points.remove (null);  
    System.out.println(points);  
}
```

What is the result?

- A. A NullPointerException is thrown at runtime.
- B. [1, 2, 4]
- C. [1, 2, 4, null ]
- D. [1, 3, 4, null ]
- E. [1, 3, 4 ]
- F. Compilation fails.

**Answer:** F

**Explanation:**

Version - JDK 1.8.0\_66

Your Code ...

```
1. public static void main (String [] args) {  
2.     ArrayList<Integer> points = new ArrayList<> () ;  
3.     points.add (1) ;  
4.     points.add (2) ;  
5.     points.add (3) ;  
6.     points.add (4) ;  
7.     points.add (null) ;  
8.     points.remove (null) ;  
9.     System.out.printIn (points) ;  
10. }
```

External Libraries ...

 Add External Library (from Maven Repo)

cs1.keyboard

Input Arguments (args of Main Method)...

Interactive mode :  OFF

Stdin Inputs...

Result...

compiled and executed in 0 second(s)

No "public class" found to execute

**NEW QUESTION 196**

Given the code fragment:

```
3. public static void main(String[] args) {  
4.     int x = 5;  
5.     while (isAvailable(x)) {  
6.         System.out.print(x);  
7.     }  
8. }  
10.  
11. public static boolean isAvailable(int x) {  
12.     return x-- > 0 ? true : false;  
13. }
```

Which modification enables the code to print 54321?

- A. Replace line 6 with System.out.print(--x);
- B. print (--x);
- C. At line7, insert x--;
- D. Replace line 6 with --x; and, at line 7, insert system.out.print(x);
- E. print (x);
- F. Replace line 12 With return (x > 0) ? false: true;

**Answer: A****NEW QUESTION 200**

Given the code fragment:

```
String[] strs = new String[2];
int idx = 0;
for (String s : strs) {
    strs[idx].concat(" element " + idx);
    idx++;
}
for (idx = 0; idx < strs.length; idx++) {
    System.out.println(strs[idx]);
}
```

What is the result?

- A. Element 0Element 1
- B. Null element 0Null element 1
- C. NullNull
- D. A NullPointerException is thrown at runtime.

Answer: C

#### NEW QUESTION 201

Given the following class declarations: Which answer fails to compile?

- A) ArrayList<Animal> myList = new ArrayList<>();
 myList.add(new Tiger());
- B) ArrayList<Hunter> myList = new ArrayList<>();
 myList.add(new Cat());
- C) ArrayList<Hunter> myList = new ArrayList<>();
 myList.add(new Tiger());
- D) ArrayList<Tiger> myList = new ArrayList<>();
 myList.add(new Cat());
- E) ArrayList<Animal> myList = new ArrayList<>();
 myList.add(new Cat());

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

Answer: E

#### NEW QUESTION 206

Given:

```
public class Test {  
    public static void main(String[] args) {  
        int x = 1;  
        int y = 0;  
        if(x++ > ++y) {  
            System.out.print("Hello ");  
        } else {  
            System.out.print("Welcome ");  
        }  
        System.out.print("Log " + x + ":" + y);  
    }  
}
```

What is the result?

- A. Hello Log 1:0
- B. Hello Log 2:1
- C. Welcome Log 2:1
- D. Welcome Log 1:0

Answer: C

#### NEW QUESTION 211

Given the code from the Greeting.java file:

```
public class Greeting {  
    public static void main(String[] args) {  
        System.out.println("Hello " + args[0]);  
    }  
}
```

Which set of commands prints Hello Duke in the console?

- A) javac Greeting  
java Greeting Duke
- B) javac Greeting.java Duke  
java Greeting
- C) javac Greeting.java  
java Greeting Duke
- D) javac Greeting.java  
java Greeting.class Duke

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: C

#### NEW QUESTION 215

Given:

MainTest.java:

```
public class MainTest {  
  
    public static void main(int[] args) {  
        System.out.println("int main " + args[0]);  
    }  
    public static void main(Object[] args) {  
        System.out.println("Object main " + args[0]);  
    }  
    public static void main(String[] args) {  
        System.out.println("String main " + args[0]);  
    }  
}
```

and commands:

```
javac MainTest.java  
java MainTest 1 2 3
```

What is the result?

- A. int main 1
- B. Object main 1
- C. String main 1
- D. Compilation fails
- E. An exception is thrown at runtime

Answer: C

#### NEW QUESTION 219

Given the code fragment:

```
public static void main(String[] args) {  
    int array[] = {10, 20, 30, 40, 50};  
    int x = array.length;  
    /* line n1 */  
}
```

Which two code fragments can be independently inserted at line n1 to enable the code to print the elements of the array in reverse order?

- A. while (x > 0) {x--;System.out.print(array[x]);}
- B. do {x--;System.out.print(array[x]);} while (x >= 0);
- C. while (x >= 0) {System.out.print(array[x]);x--;}
- D. do {System.out.print(array[x]);--x;} while (x >= 0);
- E. while (x > 0) {System.out.print(array[--x]);}

Answer: BE

#### NEW QUESTION 221

Given the code fragment:

```
int nums1[] = new int[3];  
int nums2[] = {1, 2, 3, 4, 5};  
nums1 = nums2;  
for (int x : nums1){  
    System.out.print(x + ":" );  
}
```

What is the result?

- A. 1:2:3:4:5:
- B. 1:2:3:

- C. Compilation fails.
- D. An `ArrayOutOfBoundsException` is thrown at runtime.

**Answer:** A

#### NEW QUESTION 226

Given the content of three files:

A.java:

```
public class A {  
    public void a() {}  
    int a;  
}
```

B.java:

```
public class B {  
    private int doStuff() {  
        private int x = 100;  
        return x++;  
    }  
}
```

C.java:

```
import java.io.*;  
package p1;  
class A {  
    public void main(String fileName) throws IOException {}  
}
```

Which statement is true?

- A. Only the A.java file compiles successfully.
- B. Only the B.java file compiles successfully.
- C. Only the C.java file compiles successfully.
- D. The A.java and B.java files compile successfully.
- E. The B.java and C.java files compile successfully.
- F. The A.java and C.java files compile successfully.

**Answer:** A

#### NEW QUESTION 231

Given the code fragment:

```
int wd = 0;
String days[] = {"sun", "mon", "wed", "sat"};
for (String s:days) {
    switch (s) {
        case "sat":
        case "sun":
            wd -= 1;
            break;
        case "mon":
            wd++;
        case "wed":
            wd += 2;
    }
}
System.out.println(wd);
```

What is the result?

- A. 3
- B. 4
- C. -1
- D. Compilation fails.

Answer: B

#### NEW QUESTION 234

Given the code fragment:

```
LocalDate date1 = LocalDate.now();
LocalDate date2 = LocalDate.of(2014, 6, 20);
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);
System.out.println("date1 = " + date1);
System.out.println("date2 = " + date2);
System.out.println("date3 = " + date3);
```

Assume that the system date is June 20, 2014. What is the result?

- A) date1 = 2014-06-20  
date2 = 2014-06-20  
date3 = 2014-06-20
- B) date1 = 06/20/2014  
date2 = 2014-06-20  
date3 = Jun 20, 2014
- C) Compilation fails.
- D) A DateParseException is thrown at runtime.

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: D

#### NEW QUESTION 235

Given the code fragment:

```
public static void main(String[] args) {  
    LocalDate date = LocalDate.of(2012, 01, 32);  
    date.plusDays(10);  
    System.out.println(date);  
}
```

What is the result?

- A. 2012-02-10
- B. 2012-02-11
- C. Compilation fails
- D. A DateTimeException is thrown at runtime.

**Answer:** C

**NEW QUESTION 236**

.....

## Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

### 1Z0-808 Practice Exam Features:

- \* 1Z0-808 Questions and Answers Updated Frequently
- \* 1Z0-808 Practice Questions Verified by Expert Senior Certified Staff
- \* 1Z0-808 Most Realistic Questions that Guarantee you a Pass on Your First Try
- \* 1Z0-808 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

100% Actual & Verified — Instant Download, Please Click  
[Order The 1Z0-808 Practice Test Here](#)

**NEW QUESTION 1**

Given the following classes:

```
public class Employee {  
    public int salary;  
}  
  
public class Manager extends Employee {  
    public int budget;  
}  
  
public class Director extends Manager {  
    public int stockOptions;  
}
```

And given the following main method:

```
public static void main(String[] args) {  
    Employee employee = new Employee();  
    Manager manager = new Manager();  
    Director director = new Director();  
    //line n1  
}
```

Which two options fail to compile when placed at line n1 of the main method? (Choose two.)

- A. employee.salary = 50\_000;
- B. director.salary = 80\_000;
- C. employee.budget = 200\_000;
- D. manager.budget = 1\_000\_000;
- E. manager.stockOption = 500;
- F. director.stockOptions = 1\_000;

**Answer:** CE

**NEW QUESTION 2**

Given the following main method:

```
public static void main(String[] args) {  
    int num = 5;  
    do {  
        System.out.print(num-- + " ");  
    } while(num == 0);  
}
```

What is the result?

- A. 5 4 3 2 1 0
- B. 5 4 3 2 1
- C. 4 2 1
- D. 5
- E. Nothing is printed

**Answer:** D

**NEW QUESTION 3**

Given the code fragment:

```
public static void main(String[] args) {  
    Short s1 = 200;  
    Integer s2 = 400;  
    Long s3 = (long) s1 + s2;           //line n1  
    String s4 = (String) (s3 * s2);    //line n2  
    System.out.println("Sum is " + s4);  
}
```

What is the result?

- A. Sum is 600
- B. Compilation fails at line n1.
- C. Compilation fails at line n2.
- D. A ClassCastException is thrown at line n1.
- E. A ClassCastException is thrown at line n2.

**Answer:** C

**NEW QUESTION 4**

Given:

```
class A {  
    public void test () {  
        System.out.println ("A");  
    }  
}  
class B extends A {  
    public void test () {  
        System.out.println ("B");  
    }  
}  
public class C extends A {  
    public void test () {  
        System.out.println ("C");  
    }  
  
    public static void main (String [] args) {  
        A b1 = new A ();  
        A b2 = new C ();  
  
        b1 = (A) b2;           //line n1  
        A b3 = (B) b2;         //line n2  
        b1.test ();  
        b3.test ();  
    }  
}
```

What is the result?

- A. AB
- B. AC
- C. CC
- D. A ClassCastException is thrown only at line n1.
- E. A ClassCastException is thrown only at line n2.

**Answer: B****NEW QUESTION 5**

Given the code fragment:

```
public static void main(String[] args) {  
    ArrayList<Integer> points = new ArrayList<>();  
    points.add(1);  
    points.add(2);  
    points.add(3);  
    points.add(4);  
    points.add(null);  
    points.remove(1);  
    points.remove(null);  
    System.out.println(points);  
}
```

What is the result?

- A. A NullPointerException is thrown at runtime
- B. [1, 2, 4]
- C. [1, 2, 4, null]
- D. [1, 3, 4, null]
- E. [1, 3, 4]
- F. Compilation fails.

**Answer: B****NEW QUESTION 6**

Given the code fragment:

```
int n [] [] = {{1, 3}, {2, 4}};
for (int i = n.length-1; i >= 0; i--) {
    for (int y : n[i]) {
        System.out.print (y);
    }
}
```

What is the result?

- A. 1324
- B. 2313
- C. 3142
- D. 4231

**Answer:** D

#### NEW QUESTION 7

Given:

```
class X {
    static int i;
    int j;
    public static void main(String[] args) {
        X x1 = new X();
        X x2 = new X();
        x1.i = 3;
        x1.j = 4;
        x2.i = 5;
        x2.j = 6;
        System.out.println(
            x1.i + " " +
            x1.j + " " +
            x2.i + " " +
            x2.j);
    }
}
```

What is the result?

- A. 3 4 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 4 6

**Answer:** C

#### NEW QUESTION 8

Which three statements are true about exception handling? (Choose three.)

- A. Only unchecked exceptions can be rethrown.
- B. All subclasses of the RuntimeException class are not recoverable.
- C. The parameter in a catch block is of Throwable type.
- D. All subclasses of the RuntimeException class must be caught or declared to be thrown.
- E. All subclasses of the RuntimeException class are unchecked exceptions.
- F. All subclasses of the Error class are not recoverable.

**Answer:** BCD

#### NEW QUESTION 9

Which is true about the switch statement?

- A. Its expression can evaluate to a collection of values.
- B. The break statement, at the end of each case block, is optional.
- C. Its case label literals can be changed at runtime.
- D. It must contain the default section.

**Answer:** B

#### NEW QUESTION 10

Given:

```
class Caller {  
    private void init () {  
        System.out.println("Initialized");  
    }  
  
    private void start () {  
        init();  
        System.out.println("Started");  
    }  
}  
  
public class TestCall {  
    public static void main(String[] args) {  
        Caller c = new Caller();  
        c.start(); // line n1  
        c.init(); // line n2  
    }  
}
```

What is the result?

- A. Compilation fails at line n1.
- B. InitializedStartedInitialized
- C. InitializedStarted
- D. Compilation fails at line n2.

**Answer:** D

#### NEW QUESTION 10

Given this class:

```
public class CheckingAccount {  
    public int amount;  
    //line n1  
}
```

And given this main method, located in another class:

```
public static void main(String[] args) {  
    CheckingAccount acct = new CheckingAccount();  
    //line n2  
}
```

Which three pieces of code, when inserted independently, set the value of amount to 100?

A

At line n1 insert:  
public CheckingAccount() {  
 amount = 100;  
}

B

At line n2 insert:  
this.amount = 100;

C

At line n2 insert:  
amount = 100;

D

At line n1 insert:  
public CheckingAccount() {  
 this.amount = 100;  
}

E

At line n2 insert:  
acct.amount = 100;

F

At line n1 insert:  
public CheckingAccount() {  
 acct.amount = 100;  
}

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

**Answer:** DE**NEW QUESTION 13**

Given:

```
class Test {  
    int a1;  
  
    public static void doProduct(int a) {  
        a = a * a;  
    }  
  
    public static void doString(String s) {  
        s.concat(" " + s);  
    }  
  
    public static void main(String[] args) {  
        Test item = new Test();  
        item.a1 = 11;  
        String sb = "Hello";  
        Integer i = 10;  
        doProduct(i);  
        doString(sb);  
        doProduct(item.a1);  
        System.out.println(i + " " + sb + " " + item.a1);  
    }  
}
```

What is the result?

- A. 10 Hello Hello 11
- B. 10 Hello Hello 121
- C. 100 Hello 121
- D. 100 Hello Hello 121
- E. 10 Hello 11

**Answer:** E

#### NEW QUESTION 15

Given:

```
class Test {  
    public static void main (String [] args) {  
        int numbers [ ];  
        numbers = new int [2];  
        numbers [0] = 10;  
        numbers [1] = 20;  
  
        numbers = new int [4];  
        numbers [2] = 30;  
        numbers [3] = 40;  
        for (int x : numbers) {  
            System.out.print (" " + x) ;  
        }  
    }  
}
```

What is the result?

- A. 10 20 30 40
- B. 0 0 30 40
- C. Compilation fails.
- D. An exception is thrown at runtime.

**Answer:** C

#### NEW QUESTION 17

Given the code fragment:

```
3. public static void main(String[] args) {  
4.     int x = 6;  
5.     while (isAvailable(x)) {  
6.         System.out.print(x);  
7.  
8.     }  
9. }  
10.  
11. public static boolean isAvailable(int x) {  
12.     return --x > 0 ? true : false;  
13. }
```

Which modification enables the code to print 54321?

- A. Replace line 6 with System.out.print (--x);
- B. At line 7, insert x --;
- C. Replace line 5 with while (is Available(--x)) {
- D. Replace line 12 with return (x > 0) ? false : true;

**Answer:** C

#### NEW QUESTION 19

Which statement best describes encapsulation?

- A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects.
- B. Encapsulation ensures that classes can be designed so that their methods are inheritable.
- C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.
- D. Encapsulation ensures that classes can be designed so that if a method has an argument MyType x, any subclass of MyType can be passed to that method.

**Answer:** A

**NEW QUESTION 20**

Which two statements are true? (Choose two.)

- A. Error class is unextendable.
- B. Error class is extendable.
- C. Error is a RuntimeException.
- D. Error is an Exception.
- E. Error is a Throwable.

**Answer:** BC

**NEW QUESTION 22**

Given the code fragment:

```
LocalDate date1 = LocalDate.now();
LocalDate date2 = LocalDate.of(6, 20, 2014 );
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);
System.out.println("date1 = " + date1);
System.out.println("date2 = " + date2);
System.out.println("date3 = " + date3);
```

Assume that the system date is June 20, 2014. What is the result?

A

```
date1 = 2014-06-20
date2 = 2014-06-20
date3 = 2014-06-20
```

B

```
date1 = 06/20/2014
date2 = 2014-06-20
date3 = Jun 20, 2014
```

C Compilation fails.

D An exception is thrown at runtime.

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

**NEW QUESTION 27**

Which three statements describe the object-oriented features of the Java language? (Choose three.)

- A. Objects cannot be reused.
- B. A subclass must override the methods from a superclass.
- C. Objects can share behaviors with other objects.
- D. A package must contain a main class.
- E. Object is the root class of all other objects.
- F. A main method must be declared in every class.

**Answer:** BCF

**NEW QUESTION 30**

Which statement will empty the contents of a StringBuilder variable named sb?

- A. s
- B. deleteAll();
- C. s
- D. delete(0, s
- E. size();
- F. s
- G. delete(0, s
- H. length();
- I. s
- J. removeAll();

**Answer:** C

**NEW QUESTION 33**

Given the code fragment:

```
String[] strs = {"A", "B"};
int idx = 0;
for (String s : strs) {
    strs[idx].concat(" element " + idx);
    idx++;
}
for (idx = 0; idx < strs.length; idx++) {
    System.out.println(strs[idx]);
}
```

What is the result?

- A. AB
- B. A element 0B element 1
- C. A NullPointerException is thrown at runtime.
- D. A 0B 1

**Answer:** C

#### NEW QUESTION 37

Given the code fragment:

```
int nums1[] = {1, 2, 3};
int nums2[] = {1, 2, 3, 4, 5};
nums 2 = nums 1;
for (int x : nums2){
    System.out.print(x + ":");
}
```

What is the result?

- A. 1:2:3:4:5:
- B. 1:2:3:
- C. Compilation fails.
- D. An ArrayOutOfBoundsException is thrown at runtime.

**Answer:** A

#### NEW QUESTION 40

Given the code fragment:

```
if (aVar++ < 10) {
    System.out.println(aVar + " Hello Universe!");
} else {
    System.out.println(aVar + " Hello World!");
}
```

What is the result if the integer aVar is 9?

- A. Compilation fails.
- B. 10 Hello Universe!
- C. 10 Hello World!
- D. 9 Hello World!

**Answer:** B

#### NEW QUESTION 41

Which three statements are true about the structure of a Java class? (Choose three.)

- A. A class cannot have the same name as its field.
- B. A public class must have a main method.
- C. A class can have final static methods.
- D. A class can have overloaded private constructors.
- E. Fields need to be initialized before use.
- F. Methods and fields are optional components of a class.

**Answer:** BDE

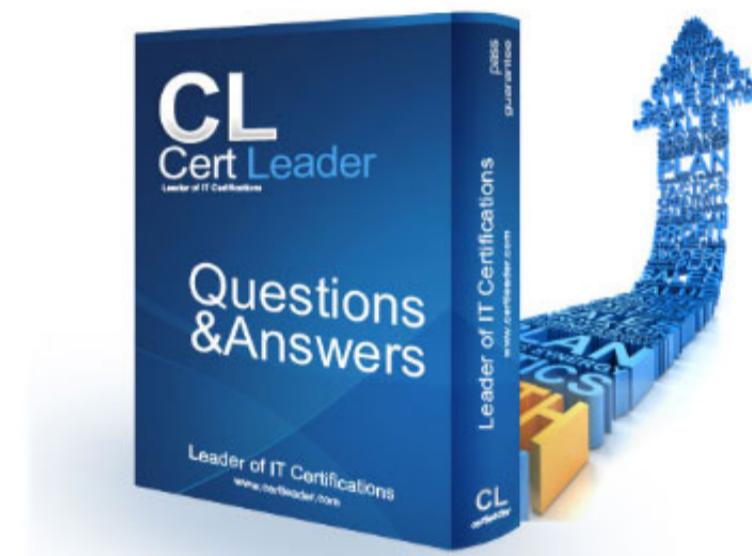
#### NEW QUESTION 44

.....

## 1z0-808 Dumps

### Java SE 8 Programmer I

<https://www.certleader.com/1z0-808-dumps.html>



**NEW QUESTION 1**

You are asked to create a method that accepts an array of integers and returns the highest value from that array.

Given the code fragment:

```
class Test{
    public static void main(String[] args) {
        int numbers[] = {12, 13, 42, 32, 15, 156, 23, 51, 12};
        int[] keys = findMax(numbers);
    }

    /* line n1 */
    int[] keys = new int[3];
    /* code goes here*/
    return keys;
}
}
```

Which method signature do you use at line n1?

- A. public int findMax (int[] numbers)
- B. static int[] findMax (int[] max)
- C. static int findMax (int[] numbers)
- D. final int findMax (int[] )

**Answer:** C

**NEW QUESTION 2**

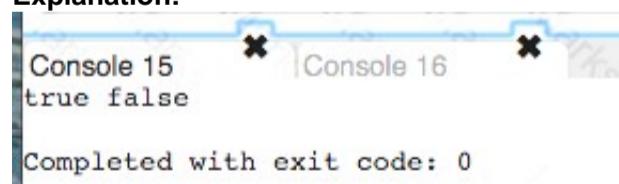
Given:

```
public class Test {
    public static void main(String[] args) {
        Test ts = new Test();
        System.out.print(isAvailable + " ");
        isAvailable= ts.doStuff();
        System.out.println(isAvailable);
    }
    public static boolean doStuff() {
        return !isAvailable;
    }
    static boolean isAvailable = true;
}
```

What is the result?

- A. Compilation fails.
- B. false true
- C. true false
- D. true true
- E. false false

**Answer:** C

**Explanation:**

```
Console 15 * Console 16 *
true false
false
Completed with exit code: 0
```

**NEW QUESTION 3**

Which statement is true about the switch statement?

- A. It must contain the default section.
- B. The break statement, at the end of each case block, is mandatory.
- C. Its case label literals can be changed at runtime.
- D. Its expression must evaluate to a single value.

**Answer:** D

**NEW QUESTION 4**

Given:

```
class A {  
    public void test () {  
        System.out.println ("A");  
    }  
}  
class B extends A {  
    public void test () {  
        System.out.println ("B");  
    }  
}  
public class C extends A {  
    public void test () {  
        System.out.println ("C");  
    }  
  
    public static void main (String [] args) {  
        A b1 = new A ();  
        A b2 = new C ();  
  
        b1 = (A) b2;           //line n1  
        A b3 = (B) b2;         //line n2  
        b1.test ();  
        b3.test ();  
    }  
}
```

What is the result?

- A. AB
- B. AC
- C. CC
- D. A ClassCastException is thrown only at line n1.
- E. A ClassCastException is thrown only at line n2.

**Answer: B****NEW QUESTION 5**

Given the code fragment:

```
public static void main(String[] args) {  
    ArrayList<Integer> points = new ArrayList<>();  
    points.add(1);  
    points.add(2);  
    points.add(3);  
    points.add(4);  
    points.add(null);  
    points.remove(1);  
    points.remove(null);  
    System.out.println(points);  
}
```

What is the result?

- A. A NullPointerException is thrown at runtime
- B. [1, 2, 4]
- C. [1, 2, 4, null]
- D. [1, 3, 4, null]
- E. [1, 3, 4]
- F. Compilation fails.

**Answer: B****NEW QUESTION 6**

Given the code from the Greeting.java file:

```
public class Greeting {  
    public static void main(String[] args) {  
        System.out.println("Hello " + args[0]);  
    }  
}
```

Which set of commands prints Hello Duke in the console?

- A) javac Greeting  
java Greeting Duke
- B) javac Greeting.java Duke  
java Greeting
- C) javac Greeting.java  
java Greeting Duke
- D) javac Greeting.java  
java Greeting.class Duke

- A. Option A  
B. Option B  
C. Option C  
D. Option D

**Answer:** C

#### NEW QUESTION 7

Given:

```
public class Fieldinit {  
    char c;  
    boolean b;  
    float f;  
    void printAll() {  
        System.out.println ("c = " + c);  
        System.out.println ("b = " + b);  
        System.out.println ("f = " + f);  
    }  
    public static void main (String [] args) {  
        FieldInit f = new FieldInit ();  
        f.printAll ();  
    }  
}
```

What is the result?

A  
c=  
b = false  
f = 0.0

B  
c= null  
b = true  
f = 0.0

C  
c=0  
b = false  
f = 0.0f

D  
c= null  
b = false  
f = 0.0F

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

#### NEW QUESTION 8

Given:

```
class Product {  
    double price;  
}  
  
public class Test {  
    public void updatePrice(Product product, double price) {  
        price = price * 2;  
        product.price = product.price + price;  
    }  
    public static void main(String[] args) {  
        Product prt = new Product();  
        prt.price = 200;  
        double newPrice = 100;  
  
        Test t = new Test();  
        t.updatePrice(prt, newPrice);  
        System.out.println(prt.price + " : " + newPrice);  
    }  
}
```

What is the result?

- A. 200.0 : 100.0
- B. 400.0 : 200.0
- C. 400.0 : 100.0
- D. Compilation fails.

**Answer:** C

#### NEW QUESTION 9

Given:

```
class X {  
    static int i;  
    int j;  
    public static void main(String[] args) {  
        X x1 = new X();  
        X x2 = new X();  
        x1.i = 3;  
        x1.j = 4;  
        x2.i = 5;  
        x2.j = 6;  
        System.out.println(  
            x1.i + " " +  
            x1.j + " " +  
            x2.i + " " +  
            x2.j);  
    }  
}
```

What is the result?

- A. 3 4 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 4 6

**Answer:** C

#### NEW QUESTION 10

Given the code fragment:

```
abstract class Toy {  
    int price;  
    // line n1  
}
```

Which three code fragments are valid at line n1?

A

```
public static void insertToy() {  
    /* code goes here */  
}
```

B

```
final Toy getToy() {  
    return new Toy();  
}
```

C

```
public void printToy();
```

D

```
public int calculatePrice() {  
    return price;  
}
```

E

```
public abstract int computeDiscount();
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** CDE

#### NEW QUESTION 10

Which is true about the switch statement?

- A. Its expression can evaluate to a collection of values.
- B. The break statement, at the end of each case block, is optional.
- C. Its case label literals can be changed at runtime.
- D. It must contain the default section.

**Answer:** B

#### NEW QUESTION 13

Given the code fragment:

```
public static void main(String[] args) {  
    LocalDate date = LocalDate.of(2012, 01, 32);  
    date.plusDays(10);  
    System.out.println(date);  
}
```

What is the result?

- A. 2012-02-10
- B. 2012-02-11
- C. Compilation fails
- D. A DateTimeException is thrown at runtime.

**Answer:** D

#### NEW QUESTION 18

Given:

```
interface I {  
    public void displayI();  
}  
abstract class C2 implements I {  
    public void displayC2() {  
        System.out.print("C2");  
    }  
}  
class C1 extends C2 {  
    public void displayI() {  
        System.out.print("C1");  
    }  
}
```

And the code fragment:

```
C2 obj1 = new C1();  
I obj2 = new C1();  
  
C2 s = (C2) obj2;  
I t = obj1;  
  
t.displayI();  
s.displayC2();
```

What is the result?

- A. C1C2
- B. C1C1
- C. Compilation fails.
- D. C2C2

**Answer:** A

**Explanation:**

The screenshot shows a Java development environment. At the top, there is a file tree for a project named 'lund' with a 'src' folder containing 'App.java'. Below the file tree is the code for 'App.java':

```
1 interface I {  
2     public void displayI();  
3 }  
4 abstract class C2 implements I {  
5     public void displayC2() {  
6         System.out.print("C2");  
7     }  
8 }  
9 class C1 extends C2 {  
10    public void displayI() {  
11        System.out.print("C1");  
12    }  
13 }  
14  
15 }  
16  
17 public class App {  
18     public static void main(String[] args) {  
19         C2 obj1 = new C1();  
20         I obj2 = new C1();  
21  
22         C2 s = (C2) obj2;  
23         I t = obj1;  
24  
25         t.displayI();  
26         s.displayC2();  
27     }  
28 }  
29 }
```

At the bottom, there are four console tabs labeled 'Console 1', 'Console 2', 'Console 3', and 'Console 4'. Console 1 shows the output: 'C1C2'. Console 2 shows 'Completed with exit code: 0'. The other two consoles are empty.

**NEW QUESTION 23**

Given this class:

```
public class CheckingAccount {  
    public int amount;  
    //line n1  
}
```

And given this main method, located in another class:

```
public static void main(String[] args) {  
    CheckingAccount acct = new CheckingAccount();  
    //line n2  
}
```

Which three pieces of code, when inserted independently, set the value of amount to 100?

A

At line n1 insert:

```
public CheckingAccount() {  
    amount = 100;  
}
```

B

At line n2 insert:

```
this.amount = 100;
```

C

At line n2 insert:

```
amount = 100;
```

D

At line n1 insert:

```
public CheckingAccount() {  
    this.amount = 100;  
}
```

E

At line n2 insert:

```
acct.amount = 100;
```

F

At line n1 insert:

```
public CheckingAccount() {  
    acct.amount = 100;  
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

**Answer:** DE

#### NEW QUESTION 26

Given:

Base.java:

```
class Base {  
    public void test(){  
        System.out.println("Base ");  
    }  
}
```

DerivedA.java:

```
class DerivedA extends Base {  
    public void test(){  
        System.out.println("DerivedA ");  
    }  
}
```

DerivedB.java:

```
class DerivedB extends DerivedA {  
    public void test(){  
        System.out.println("DerivedB ");  
    }  
    public static void main(String[] args) {  
        Base b1 = new DerivedB();  
        Base b2 = new DerivedA();  
        Base b3 = new DerivedB();  
        Base b4 = b3;  
        b1 = (Base) b2;  
        b1.test();  
        b4.test();  
    }  
}
```

What is the result?

- A. BaseDerivedA
- B. BaseDerivedB
- C. DerivedBDerivedB
- D. DerivedBDerivedA
- E. A ClassCastException is thrown at runtime.

**Answer:** D

#### NEW QUESTION 29

Given the code snippet from a compiled Java source file:

```
public class MyFile  
{  
    public static void main (String[] args)  
    {  
        String arg1 = args[1];  
        String arg2 = args[2];  
        String arg3 = args[3];  
        System.out.println("Arg is " + arg3);  
    }  
}
```

Which command-line arguments should you pass to the program to obtain the following output? Arg is 2

- A. java MyFile 1 3 2 2
- B. java MyFile 2 2 2
- C. java MyFile 1 2 2 3 4
- D. java MyFile 0 1 2 3

**Answer:** A

#### NEW QUESTION 30

What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?

- A. Encapsulation
- B. Inheritance
- C. Abstraction
- D. Instantiation
- E. Polymorphism

**Answer:** A

**Explanation:**

Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

**NEW QUESTION 32**

Given:

```
class Caller {  
    private void init () {  
        System.out.println("Initialized");  
    }  
  
    private void start () {  
        init();  
        System.out.println("Started");  
    }  
}  
  
public class TestCall {  
    public static void main(String[] args) {  
        Caller c = new Caller();  
        c.start();  
        c.init();  
    }  
}
```

What is the result?

- A. An exception is thrown at runtime.
- B. InitializedStartedInitialized
- C. InitializedStarted
- D. Compilation fails.

**Answer:** D

**NEW QUESTION 36**

Given the code fragment:

```
3. public static void main(String[] args) {  
4.     int x = 6;  
5.     while (isAvailable(x)) {  
6.         System.out.print(x);  
7.     }  
9. }  
10.  
11. public static boolean isAvailable(int x) {  
12.     return --x > 0 ? true : false;  
13. }
```

Which modification enables the code to print 54321?

- A. Replace line 6 with System.out.print (--x);
- B. At line 7, insert x --;
- C. Replace line 5 with while (is Available(--x)) {
- D. Replace line 12 with return (x > 0) ? false : true;

**Answer:** C

**NEW QUESTION 39**

Which statement will empty the contents of a StringBuilder variable named sb?

- A. s
- B. deleteAll ();
- C. s
- D. delete (0, s
- E. size () ;
- F. s
- G. delete (0, s
- H. length () ;
- I. s
- J. removeAll ();

**Answer:** C

**NEW QUESTION 40**

Given the code fragment:

```
int nums1[] = {1, 2, 3};  
int nums2[] = {1, 2, 3, 4, 5};  
nums 2 = nums 1;  
for (int x : nums2){  
    System.out.print(x + ":" );  
}
```

What is the result?

- A. 1:2:3:4:5:
- B. 1:2:3:
- C. Compilation fails.
- D. An ArrayOutOfBoundsException is thrown at runtime.

**Answer:** A

**NEW QUESTION 43**

Given:

```
class Vehicle {  
    int x;  
    Vehicle(){  
        this(10); // line n1  
    }  
    Vehicle(int x) {  
        this.x = x;  
    }  
}  
  
class Car extends Vehicle {  
    int y;  
    Car() {  
        super();  
        this(20); // line n2  
    }  
    Car(int y) {  
        this.y = y;  
    }  
    public String toString() {  
        return super.x + ":" + this.y;  
    }  
}
```

And given the code fragment:

And given the code fragment:

```
Vehicle y = new Car();  
System.out.println(y);
```

What is the result?

- A. 10:20
- B. 0:20
- C. Compilation fails at line n1
- D. Compilation fails at line n2

**Answer:** D

**NEW QUESTION 44**

Which three statements are true about the structure of a Java class? (Choose three.)

- A. A public class must have a main method.
- B. A class can have only one private constructors.
- C. A method can have the same name as a field.
- D. A class can have overloaded static methods.
- E. The methods are mandatory components of a class.
- F. The fields need not be initialized before use.

**Answer:** ACE

**NEW QUESTION 45**

Which three statements are true about the structure of a Java class? (Choose three.)

- A. A class cannot have the same name as its field.

- B. A public class must have a main method.
- C. A class can have final static methods.
- D. A class can have overloaded private constructors.
- E. Fields need to be initialized before use.
- F. Methods and fields are optional components of a class.

**Answer:** BDE

#### **NEW QUESTION 46**

Given:

```
public class App {  
    public static void main(String[] args) {  
        int i = 10;  
        int j = 20;  
        int k =(j += i)/ 5;  
        System.out.print(i + " : " + j + " : " + k);  
    }  
}
```

What is the result?

- A. 10 : 30 : 6
- B. 10 : 22 : 22
- C. 10 : 22 : 20
- D. 10 : 22 : 6

**Answer:** A

#### **NEW QUESTION 48**

.....

## Thank You for Trying Our Product

\* **100% Pass or Money Back**

All our products come with a 90-day Money Back Guarantee.

\* **One year free update**

You can enjoy free update one year. 24x7 online support.

\* **Trusted by Millions**

We currently serve more than 30,000,000 customers.

\* **Shop Securely**

All transactions are protected by VeriSign!

### 100% Pass Your 1z0-808 Exam with Our Prep Materials Via below:

<https://www.certleader.com/1z0-808-dumps.html>

# Oracle

## Exam Questions 1Z0-808

Java SE 8 Programmer I



**NEW QUESTION 1**

Given:

Base.java:

```
class Base {  
    public void test(){  
        System.out.println("Base ");  
    }  
}
```

DerivedA.java:

```
class DerivedA extends Base {  
    public void test(){  
        System.out.println("DerivedA ");  
    }  
}
```

DerivedB.java:

```
class DerivedB extends DerivedA {  
    public void test(){  
        System.out.println("DerivedB ");  
    }  
    public static void main(String[] args) {  
        Base b1 = new DerivedB();  
        Base b2 = new DerivedA();  
        Base b3 = new DerivedB();  
        b1 = (Base) b3;  
        Base b4 = (DerivedA) b3;  
        b1.test();  
        b4.test();  
    }  
}
```

What is the result?

- A. BaseDerivedA
- B. BaseDerivedB
- C. DerivedBDerivedB
- D. DerivedBDerivedA
- E. A classcast Exception is thrown at runtime.

**Answer: C****NEW QUESTION 2**

Given the code fragment:

```
3. public static void main(String[] args) {  
4.     int iVar = 100;  
5.     float fVar = 100.100f;  
6.     double dVar = 123;  
7.     iVar = fVar;  
8.     fVar = iVar;  
9.     dVar = fVar;  
10.    fVar = dVar;  
11.    dVar = iVar;  
12.    iVar = dVar;  
13. }
```

Which three lines fail to compile?

- A. Line 7
- B. Line 8
- C. Line 9
- D. Line 10
- E. Line 11
- F. Line 12

**Answer:** ADF

#### NEW QUESTION 3

You are asked to create a method that accepts an array of integers and returns the highest value from that array.

Given the code fragment:

```
class Test {  
    public static void main (String [] args) {  
        int numbers [] = {12, 13, 42, 32, 15, 156, 23, 51, 12};  
        int max = findMax (numbers);  
    }  
    /*line n1 */ {  
        int max = 0;  
        /* code goes here*/  
        return max;  
    }  
}
```

Which method signature do you use at line n1?

- A. public int findMax (int [] numbers)
- B. static int[] findMax (int max)
- C. static int findMax (int [] numbers)
- D. final int findMax (int [] )

**Answer:** A

#### NEW QUESTION 4

Given:

```
public class SumTest {  
  
    public static void doSum(Integer x, Integer y) {  
        System.out.println("Integer sum is " + (x + y));  
    }  
  
    public static void doSum(double x, double y) {  
        System.out.println("double sum is " + (x + y));  
    }  
  
    public static void doSum(float x, float y) {  
        System.out.println("float sum is " + (x + y));  
    }  
  
    public static void doSum(int x, int y) {  
        System.out.println("int sum is " + (x + y));  
    }  
  
    public static void main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

What is the result?

- A) int sum is 30  
float sum is 30.0
- B) int sum is 30  
double sum is 30
- C) Integer sum is 30  
double sum is 30.0
- D) Integer sum is 30  
float sum is 30.0

- A. Option A  
B. Option B  
C. Option C  
D. Option D

**Answer:** B

#### NEW QUESTION 5

Given the code fragment:

```
LocalDate date1 = LocalDate.now();  
LocalDate date2 = LocalDate.of(2014, 6, 20);  
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);  
System.out.println("date1 = " + date1);  
System.out.println("date2 = " + date2);  
System.out.println("date3 = " + date3);
```

Assume that the system date is June 20, 2014. What is the result?

- A. Compilation fails.

- B. A DateParseException is thrown at runtime
- C. Date1 = 2014-05-20Date2 = 2014-05-20Date3 = 2014-05-20
- D. date1 = 06/20/2014 date2 = 2014-06-20date3 = Jun 20, 2014

Answer: C

#### NEW QUESTION 6

Given the code fragments:

```
Interface Exportable {  
    Void export();  
}  
  
class Tool implements Exportable {  
    protected void export () {           //line n1  
        System.out.println("Tool::export");  
    }  
}  
  
class ReportTool extends Tool implements Exportable {  
  
    public void export() {             //line n2  
        System.out.println("RTool::export");  
    }  
  
    public static void main(String[] args) {  
        Tool aTool = new ReportTool();  
        Tool bTool = new Tool();  
        callExport(aTool);  
        callExport(bTool);  
    }  
  
    public static void callExport (Exportable ex) {  
        ex.export();  
    }  
}
```

What is the result?

- A. Compilation fails only at line n2.
- B. RTool::exportTool::export
- C. Tool::exportTool::export
- D. Compilation fails only at line n1.
- E. Compilation fails at both line n1 and line n2.

Answer: E

#### NEW QUESTION 7

Given the code fragment:

```
public static void main(String[] args) {  
    StringBuilder sb = new StringBuilder(5);  
    String s = "";  
  
    if (sb.equals(s)) {  
        System.out.println("Match 1");  
    } else if (sb.toString().equals(s.toString())) {  
        System.out.println("Match 2");  
    } else {  
        System.out.println("No Match");  
    }  
}
```

What is the result?

- A. Match 1
- B. Match 2
- C. No Match
- D. A NullPointerException is thrown at runtime.

**Answer:** B

#### **NEW QUESTION 8**

Given the code fragment:

```
public static void main(String[] args) {  
    String str = " ";  
    str.trim();  
    System.out.println(str.equals("") + " " + str.isEmpty());  
}
```

What is the result?

- A. true true
- B. true false
- C. false false
- D. false true

**Answer:** C

#### **NEW QUESTION 9**

Given:

```
public class Vowel {  
    private char var;  
    public static void main(String[] args) {  
        char var1 = 'a';  
        char var2 = var1;  
        var2 = 'e';  
  
        Vowel obj1 = new Vowel ();  
        Vowel obj2 = obj1;  
        obj1.var = 'i';  
        obj2.var = 'o';  
  
        System.out.println(var1 + ", " +var2);  
        System.out.print(obj1.var + ", " +obj2.var);  
    }  
}
```

What is the result?

- A. e, ei, o
- B. a, ei, o
- C. a, eo, o
- D. e, eo, o

**Answer:** A

#### NEW QUESTION 10

Given:

```
class Equal {  
    public static void main (String [] args) {  
        String str1 = "Java";  
        String [] str2 = { "J", "a", "v", "a"};  
        String str3 = "";  
        for (String str : str2) {  
            str3 = str3+str;  
        }  
        boolean b1 = (str1== str3);  
        boolean b2 = (str1.equals (str3));  
        System.out.print (b1+", "+b2);  
    }  
}
```

What is the result?

- A. false, false
- B. false, true
- C. true, false
- D. true, true

**Answer:** B

#### NEW QUESTION 10

Given:

```
public class Fieldinit {  
    char c;  
    boolean b;  
    float f;  
    void printAll() {  
        System.out.println ("c = " + c);  
        System.out.println ("b = " + b);  
        System.out.println ("f = " + f);  
    }  
    public static void main (String [] args) {  
        FieldInit f = new FieldInit ();  
        f.printAll ();  
    }  
}
```

What is the result?

- A. c=b = falsef = 0.0
- B. c= nullb = truef = 0.0
- C. c=0b = falsef = 0.0f
- D. c= nullb = falsef = 0.0F

**Answer:** C

#### NEW QUESTION 15

Which three statements describe the object-oriented features of the Java language?

- A. Objects cannot be reused.
- B. A subclass can inherit from a superclass.
- C. Objects can share behaviors with other objects.
- D. A package must contain more than one class.
- E. Object is the root class of all other objects.
- F. A main method must be declared in every class.

**Answer:** BCF

#### NEW QUESTION 19

Given the code fragment:

```
public class Test {  
    static int count = 0;  
    int i = 0;  
  
    public void changeCount() {  
        while (i < 5) {  
            i++;  
            count++;  
        }  
    }  
  
    public static void main(String[] args) {  
        Test check1 = new Test();  
        Test check2 = new Test();  
        check1.changeCount();  
        check2.changeCount();  
        System.out.print(check1.count + " : " + check2.count);  
    }  
}
```

What is the result?

- A. 10 : 10
- B. 5 : 5
- C. 5 : 10
- D. Compilation fails

**Answer:** A

#### NEW QUESTION 24

Given:

```
public static void main(String[] args) {  
    String ta = "A ";  
    ta = ta.concat("B ");  
    String tb = "C ";  
    ta = ta.concat(tb);  
    ta.replace('C', 'D');  
    ta = ta.concat(tb);  
    System.out.println(ta);  
}
```

What is the result?

- A. A B C D
- B. A C D
- C. A B C C
- D. A B D
- E. A B D C

**Answer:** E

#### NEW QUESTION 29

Given:

```
public class SumTest {  
  
    public static void doSum(Integer x, Integer y) {  
        System.out.println("Integer sum is " + (x + y));  
    }  
  
    public static void doSum(double x, double y) {  
        System.out.println("double sum is " + (x + y));  
    }  
  
    public static void doSum(float x, float y) {  
        System.out.println("float sum is " + (x + y));  
    }  
  
    public static void doSum(int x, int y) {  
        System.out.println("int sum is " + (x + y));  
    }  
  
    public static void main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

What is the result?

- A. int sum is 30float sum is 30.0
- B. int sum is 30double sum is 30.0
- C. integer sum is 30double sum is 30.0
- D. integer sum is 30float sum is 30.0

**Answer:** D

#### NEW QUESTION 32

Which statement is true about Java byte code?

- A. It can run on any platform.
- B. It can run on any platform only if it was compiled for that platform.
- C. It can run on any platform that has the Java Runtime Environment.
- D. It can run on any platform that has a Java compiler.
- E. It can run on any platform only if that platform has both the Java Runtime Environment and a Java compiler.

**Answer:** ACDE

#### NEW QUESTION 36

Given the code fragment:

```
public static void main(String[] args) {  
    List<String> names = new ArrayList<>();  
    names.add("Robb");  
    names.add("Bran");  
    names.add("Rick");  
    names.add("Bran");  
  
    if (names.remove("Bran")) {  
        names.remove("Jon");  
    }  
    System.out.println(names);  
}
```

What is the result?

- A. [Robb, Rick, Bran]
- B. [Robb, Rick]
- C. [Robb, Bran, Rick, Bran]
- D. An exception is thrown at runtime.

**Answer:** A

#### NEW QUESTION 41

Given the code fragment:

```
public static void main (String [] args) {  
    String names [] = {"Thomas", "Peter", "Joseph");  
    String pws [] = new String [3];  
    int idx = 0;  
    try {  
        for (String n: names) {  
            pws [idx] = n.substring (2, 6);  
            idx++;  
        }  
    }  
    catch (Exception e) {  
        System.out.println ("Invalid Name");  
    }  
    for (String p: pws) {  
        System.out.println (p);  
    }  
}
```

What is the result?

- A. Invalid Name
- B. Invalid Nameomas
- C. Invalid Name omas null null
- D. omasterseph

**Answer:** C

#### NEW QUESTION 43

Given:

```
class Test {  
    public static void main (String [] args) {  
        int numbers [ ];  
        numbers = new int [2];  
        numbers [0] = 10;  
        numbers [1] = 20;  
  
        numbers = new int [4];  
        numbers [2] = 30;  
        numbers [3] = 40;  
        for (int x : numbers) {  
            System.out.print (" " + x) ;  
        }  
    }  
}
```

What is the result?

- A. 10 20 30 40
- B. 0 0 30 40
- C. Compilation fails.
- D. An exception is thrown at runtime.

**Answer:** B

#### NEW QUESTION 48

Given the following classes:

```
public class Employee {  
    public int salary;  
}  
  
public class Manager extends Employee {  
    public int budget;  
}  
  
public class Director extends Manager {  
    public int stockOptions;  
}
```

And given the following main method:

```
public static void main(String[] args) {  
    Employee employee = new Employee();  
    Manager manager = new Manager();  
    Director director = new Director();  
    //line n1  
}
```

Which two options fail to compile when placed at line n1 of the main method?

- A. employee.salary = 50\_000;

- B. director.salary = 80\_000;
- C. employee.budget = 200\_000;
- D. manager.budget = 1\_000\_000;
- E. manager.stockOption = 500;
- F. director.stockOptions = 1\_000;

Answer: CE

**NEW QUESTION 53**

You are developing a banking module. You have developed a class named ccMask that has a maskCC method. Given the code fragment:

```
class CCmask {
    public static String maskCC(String creditCard) {
        String x = "XXXX-XXXX-XXXX-";
        //line n1
    }

    public static void main(String[] args) {
        System.out.println(maskCC("1234-5678-9101-1121"));
    }
}
```

You must ensure that the maskCC method returns a string that hides all digits of the credit card number except the four last digits (and the hyphens that separate each group of four digits).

Which two code fragments should you use at line n1, independently, to achieve this requirement?

- A) `StringBuilder sb = new StringBuilder(creditCard);
 sb.substring(15, 19);
 return x + sb;`
- B) `return x + creditCard.substring(15, 19);`
- C) `StringBuilder sb = new StringBuilder(x);
 sb.append(creditCard, 15, 19);
 return sb.toString();`
- D) `StringBuilder sb = new StringBuilder(creditCard);
 StringBuilder s = sb.insert(0, x);
 return s.toString();`

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: BC

**NEW QUESTION 54**

Given the code fragment:

```
public static void main(String[] args) {
    ArrayList myList = new ArrayList();
    String[] myArray;
    try {
        while (true) {
            myList.add("My String");
        }
    } catch (RuntimeException re) {
        System.out.println("Caught a RuntimeException");
    } catch (Exception e) {
        System.out.println("Caught an Exception");
    }
    System.out.println("Ready to use");
}
```

What is the result?

- A. Execution terminates in the first catch statement, and caught a RuntimeException is printed to the console.
- B. Execution terminates in the second catch statement, and caught an Exception is printed to the console.
- C. A runtime error is thrown in the thread "main".
- D. Execution completes normally, and Ready to use is printed to the console.
- E. The code fails to compile because a throws keyword is required.

**Answer:** C

#### NEW QUESTION 57

Which statement best describes encapsulation?

- A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects.
- B. Encapsulation ensures that classes can be designed so that their methods are inheritable.
- C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.
- D. Encapsulation ensures that classes can be designed so that if a method has an argument MyType x, any subclass of MyType can be passed to that method.

**Answer:** A

#### NEW QUESTION 62

Given the code fragment:

```
public class App {
    public static void main(String[] args) {
        String str1 = "Java";
        String str2 = new String("java");
        //line n1
        {
            System.out.println("Equal");
        } else {
            System.out.println("Not Equal");
        }
    }
}
```

Which code fragment, when inserted at line n1, enables the App class to print Equal?

- A) String str3 = str2;  
    if (str1 == str3)
- B) if (str1.equalsIgnoreCase(str2))
- C) String str3 = str2;  
    if (str1.equals(str3))
- D) if (str1.toLowerCase() == str2.toLowerCase())

- A. Option A  
B. Option B  
C. Option C  
D. Option D

**Answer:** A

**NEW QUESTION 67**

Given:

```
String stuff = "TV";
String res = null;

if (stuff.equals ("TV")) {
    res = "Walter";
} else if (stuff.equals ("Movie") ) {
    res= "White";
} else {
    res= "No Result";
}
```

Which code fragment can replace the if block?

- A. stuff.equals ("TV") ? res= "Walter" : stuff.equals ("Movie") ? res = "White" : res = "No Result";  
B. res = stuff.equals ("TV") ? "Walter" else stuff.equals ("Movie")? "White" : "No Result";  
C. res = stuff.equals ("TV") ? stuff.equals ("Movie")? "Walter" : "White" : "No Result";  
D. res = stuff.equals ("TV")? "Walter" : stuff.equals ("Movie")? "White" : "No Result";

**Answer:** B

**NEW QUESTION 68**

Given:

```
interface Readable {  
    public void readBook();  
    public void setBookMark();  
}  
  
abstract class Book implements Readable { // line n1  
    public void readBook() {}  
    // line n2  
}  
  
class EBook extends Book { // line n3  
    public void readBook() {}  
    // line n4  
}
```

And given the code fragment: Book book1 = new EBook(); Book1.readBook();  
Which option enables the code to compile?

- A. Replace the code fragment at line n3 with:  
abstract class EBook extends Book {
- B. Replace the code fragment at line n1 with:  
class Book implements Readable {
- C. At line n2 insert:  
public abstract void setBookMark();
- D. At line n4 insert:  
public void setBookMark() {}

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

#### NEW QUESTION 71

Given the code snippet from a compiled Java source file:

```
public class MyFile  
{  
    public static void main (String[] args)  
    {  
        String arg1 = args[1];  
        String arg2 = args[2];  
        String arg3 = args[3];  
        System.out.println("Arg is " + arg3);  
    }  
}
```

Which command-line arguments should you pass to the program to obtain the following output? Arg is 2

- A. java MyFile 1 3 2 2
- B. java MyFile 2 2 2
- C. java MyFile 1 2 2 3 4

D. java MyFile 0 1 2 3

Answer: A

#### NEW QUESTION 76

Given:

```
public class App {  
  
    String myStr = "7007";  
  
    public void doStuff(String str) {  
        int myNum = 0;  
        try {  
            String myStr = str;  
            myNum = Integer.parseInt(myStr);  
        } catch (NumberFormatException ne) {  
            System.err.println("Error");  
        }  
        System.out.println(  
            "myStr: " + myStr + ", myNum: " + myNum);  
    }  
  
    public static void main(String[] args) {  
        App obj = new App();  
        obj.doStuff("9009");  
    }  
}
```

What is the result?

- A. myStr: 9009, myNum: 9009
- B. myStr: 7007, myNum: 7007
- C. myStr: 7007, myNum: 9009
- D. Compilation fails

Answer: C

#### NEW QUESTION 78

Given the code fragment:

```
abstract class Planet {  
    protected void revolve() { //line n1  
    }  
  
    abstract void rotate(); //line n2  
}  
  
class Earth extends Planet {  
    void revolve() { //line n3  
    }  
  
    protected void rotate() { //line n4  
    }  
}
```

Which two modifications, made independently, enable the code to compile?

- A. Make the method at line n1 public.
- B. Make the method at line n2 public.

- C. Make the method at line n3 public.
- D. Make the method at line n3 protected.
- E. Make the method at line n4 public.

Answer: BC

#### NEW QUESTION 82

Given the following code:

```
int[] intArr = {15, 30, 45, 60, 75};  
intArr[2] = intArr[4];  
intArr[4] = 90;
```

What are the values of each element in intArr after this code has executed?

- A. 15, 60, 45, 90, 75
- B. 15, 90, 45, 90, 75
- C. 15, 30, 75, 60, 90
- D. 15, 30, 90, 60, 90
- E. 15, 4, 45, 60, 90

Answer: C

#### NEW QUESTION 84

Given the code fragment:

```
if (aVar++ < 10) {  
    System.out.println(aVar + " Hello World!");  
} else {  
    System.out.println(aVar + " Hello Universe!");  
}
```

What is the result if the integer aVar is 9?

- A. Compilation fails.
- B. 10 Hello Universe!
- C. 10 Hello World!
- D. 9 Hello World!

Answer: C

#### NEW QUESTION 85

You are asked to develop a program for a shopping application, and you are given the following information: Which definition of the Toy class adds a valid layer of abstraction to the class hierarchy?

- A) 

```
public abstract class Toy{  
    public abstract int calculatePrice(Toy t);  
    public void printToy(Toy t) { /* code goes here */ }  
}
```
- B) 

```
public abstract class Toy {  
    public int calculatePrice(Toy t) ;  
    public void printToy(Toy t) ;  
}
```
- C) 

```
public abstract class Toy {  
    public int calculatePrice(Toy t);  
    public final void printToy(Toy t){ /* code goes here */ }  
}
```
- D) 

```
public abstract class Toy {  
    public abstract int calculatePrice(Toy t) { /* code goes here */ }  
    public abstract void printToy(Toy t) { /* code goes here */ }  
}
```

- A. Option A
- B. Option B

- C. Option C
- D. Option D

**Answer:** A

#### NEW QUESTION 87

Given:

```
public class Test {  
    public static void main(String[] args) {  
        Test ts = new Test();  
        System.out.print(isAvailable + " ");  
        isAvailable= ts.doStuff();  
        System.out.println(isAvailable);  
    }  
    public static boolean doStuff() {  
        return !isAvailable;  
    }  
    static boolean isAvailable = false;  
}
```

What is the result?

- A. Compilation fails.
- B. false true
- C. true false
- D. true true
- E. false false

**Answer:** B

#### NEW QUESTION 91

Which two statements are true?

- A. Error class is unextendable.
- B. Error class is extendable.
- C. Error is a RuntimeException.
- D. Error is an Exception.
- E. Error is a Throwable.

**Answer:** BC

#### NEW QUESTION 92

Which three statements are true about exception handling?

- A. Only unchecked exceptions can be rethrown.
- B. All subclasses of the RuntimeException class are recoverable.
- C. The parameter in a catch block is of Throwable type.
- D. All subclasses of the RuntimeException class must be caught or declared to be thrown.
- E. All subclasses of the Exception class except the RuntimeException class are checked exceptions.
- F. All subclasses of the Error class are checked exceptions and are recoverable.

**Answer:** BCE

#### NEW QUESTION 94

Given:

```
public class MarkList {  
    int num;  
    public static void graceMarks(MarkList obj4) {  
        obj4.num += 10;  
    }  
    public static void main(String[] args) {  
        MarkList obj1 = new MarkList();  
        MarkList obj2 = obj1;  
        MarkList obj3 = null;  
        obj2.num = 60;  
        graceMarks(obj2);  
    }  
}
```

How many MarkList instances are created in memory at runtime?

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

Answer: A

**NEW QUESTION 95**

Given the code fragment:

```
public static void main(String[] args) {  
    int ii = 0;  
    int jj = 7;  
    for (ii = 0; ii < jj - 1; ii = ii + 2) {  
        System.out.print(ii + " ");  
    }  
}
```

What is the result?

- A. 2 4
- B. 0 2 4 6
- C. 0 2 4
- D. Compilation fails

Answer: C

**NEW QUESTION 98**

Given the code fragment:

```
if (aVar++ < 10) {  
    System.out.println(aVar + " Hello World!");  
} else {  
    System.out.println(aVar + " Hello Universe!");  
}
```

What is the result if the integer aVar is 9?

- A. 10 Hello World!
- B. Hello Universe!
- C. Hello World!
- D. Compilation fails.

Answer: A

**NEW QUESTION 102**

Given the code fragment:

```
4. class X {  
5.     public void printFileContent () {  
6.         /* code goes here */  
7.         throw new IOException ();  
8.     }  
9. }  
10. public class Test {.  
11.     public static void main (String [] args) {  
12.         X xobj = new X ();  
13.         xobj.printFileContent ();  
14.     }  
15. }
```

Which two modifications should you make so that the code compiles successfully?

- A. At line 14, insert `throw new IOException ();`
- B. Replace line 5 with `public void printFileContent () throws IOException {`
- C. Replace line 11 with `public static void main (String [] args) throws Exception {`
- D. Replace line 13 with:

```
try {  
    xobj.printFileContent ();  
}  
catch (Exception e) {}  
catch (IOException e) {}
```

- E. Replace line 7 with `throw IOException ("Exception raised");`

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** E

#### NEW QUESTION 104

Given:

```
class Vehicle {  
    int x;  
    Vehicle() {  
        this(10); // line n1  
    }  
    Vehicle(int x) {  
        this.x = x;  
    }  
}  
  
class Car extends Vehicle {  
    int y;  
    Car() {  
        super();  
        this(20); // line n2  
    }  
    Car(int y) {  
        this.y = y;  
    }  
    public String toString() {  
        return super.x + ":" + this.y;  
    }  
}
```

And given the code fragment:

And given the code fragment:

```
Vehicle y = new Car();  
System.out.println(y);
```

What is the result?

- A. 10:20
- B. 0:20
- C. Compilation fails at line n1
- D. Compilation fails at line n2

**Answer: D**

#### NEW QUESTION 109

Given:

```
public class App {  
    int count;  
    public static void displayMsg () {  
        count++; // line n1  
        System.out.println ("Welcome "+"Visit Count: "+count); // line n2  
    }  
    public static void main (String [] args) {  
        App.displayMsg (); // line n3  
        App.displayMsg (); // line n4  
    }  
}
```

What is the result?

- A. Compilation fails at line n3 and line n4.
- B. Compilation fails at line n1 and line n2.
- C. Welcome Visit Count:1Welcome Visit Count: 2
- D. Welcome Visit Count:1Welcome Visit Count: 2

**Answer:** B

#### NEW QUESTION 112

Given the code fragment from three files:

SalesMan.java:

```
package sales;  
public class SalesMan { }
```

Product.java:

```
package sales.products;  
public class Product { }
```

Market.java:

```
1. package market;  
2. // insert code here  
3. public class USMarket {  
4.     SalesMan sm;  
5.     Product p;  
6. }
```

Which code fragment, when inserted at line 2, enables the code to compile?

- A) import sales.\*;
- B) import java.sales.products.\*;
- C) import sales;  
    import sales.products;
- D) import sales.\*;  
    import products.\*;
- E) import sales.\*;  
    import sales.products.\*;

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** E

#### NEW QUESTION 117

Given:

```
class C2 {  
    public void displayC2() {  
        System.out.print("C2");  
    }  
}  
interface I {  
    public void displayI();  
}  
class C1 extends C2 implements I {  
    public void displayI() {  
        System.out.print("C1");  
    }  
}
```

And given the code fragment:

```
C2 obj1 = new C1();  
I obj2 = new C1();  
  
C2 s = obj2;  
I t = obj1;  
  
t.displayI();  
s.displayC2()
```

What is the result?

- A. C2C2
- B. C1C2
- C. C1C1
- D. Compilation fails

**Answer:** A

#### NEW QUESTION 118

Which three are advantages of the Java exception mechanism?

- A. Improves the program structure because the error handling code is separated from the normal program function
- B. Provides a set of standard exceptions that covers all the possible errors
- C. Improves the program structure because the programmer can choose where to handle exceptions
- D. Improves the program structure because exceptions must be handled in the method in which they occurred
- E. Allows the creation of new exceptions that are tailored to the particular program being created

Answer: ACD

#### NEW QUESTION 121

Given:

```
class X {  
    static int i;  
    int j;  
    public static void main(String[] args) {  
        X x1 = new X();  
        X x2 = new X();  
        x1.i = 3;  
        x1.j = 4;  
        x2.i = 5;  
        x2.j = 6;  
        System.out.println(  
            x1.i + " " +  
            x1.j + " " +  
            x2.i + " " +  
            x2.j);  
    }  
}
```

What is the result?

- A. 3 4 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 4 6

Answer: C

#### NEW QUESTION 126

Given the following code:

```
public static void main(String[] args){  
    String[] planets = {"Mercury", "Venus", "Earth", "Mars"};  
  
    System.out.println(planets.length);  
    System.out.println(planets[1].length());  
}
```

What is the output?

- A. 44
- B. 35
- C. 47
- D. 54
- E. 45
- F. 421

Answer: E

#### NEW QUESTION 127

Given:

```
public class Vowel {  
    private char var;  
    public static void main(String[] args) {  
        char var1 = 'a';  
        char var2 = var1;  
        var2 = 'e';  
  
        Vowel obj1 = new Vowel ();  
        Vowel obj2 = obj1;  
        obj1.var = 'i';  
        obj2.var = 'o';  
  
        System.out.println(var1 + ", " +var2);  
        System.out.print(obj1.var + ", " +obj2.var);  
    }  
}
```

What is the result?

- A. a, oi, o
- B. a, oo, o
- C. o, oi, o
- D. o, oo, o

**Answer:** B

#### NEW QUESTION 130

Given the code fragment:

```
public static void main(String[] args) {  
    String[] arr = {"A", "B", "C", "D"};  
    for (int i = 0; i < arr.length; i++) {  
        System.out.print(arr[i] + " ");  
        if (arr[i].equals("C")) {  
            continue;  
        }  
        System.out.println("Work done");  
        break;  
    }  
}
```

What is the result?

- A. A B C Work done
- B. A B C D Work done
- C. A Work done
- D. Compilation fails

**Answer:** C

#### NEW QUESTION 135

Given the code fragment:

```
public class Employee {  
    String name;  
    boolean contract;  
    double salary;  
    Employee() {  
        // line n1  
    }  
    public String toString() {  
        return name + ":" + contract + ":" + salary;  
    }  
    public static void main(String[] args) {  
        Employee e = new Employee();  
        // line n2  
        System.out.print(e);  
    }  
}
```

Which two modifications, when made independently, enable the code to print joe:true: 100.0?

- A) Replace line n2 with:

```
e.name = "Joe";  
e.contract = true;  
e.salary = 100;
```

- B) Replace line n2 with:

```
this.name = "Joe";  
this.contract = true;  
this.salary = 100;
```

- C) Replace line n1 with:

```
this.name = new String("Joe");  
this.contract = new Boolean(true);  
this.salary = new Double(100);
```

- D) Replace line n1 with:

```
name = "Joe";  
contract = TRUE;  
salary = 100.0f;
```

- E) Replace line n1 with:

```
this("Joe", true, 100);
```

- A. Option A  
B. Option B  
C. Option C  
D. Option D  
E. Option E

**Answer:** AC

#### NEW QUESTION 140

Given:

```
interface Downloadable {
    public void download();
}

interface Readable extends Downloadable { // line n1
    public void readBook();
}

abstract class Book implements Readable { // line n2
    public void readBook() {
        System.out.println("Read Book");
    }
}

class EBook extends Book { // line n3
    public void readBook() {
        System.out.println("Read E-Book");
    }
}
```

And given the code fragment:

```
Book book1 = new EBook();
book1.readBook();
```

What is the result?

- A. Compilation fails at line n2.
- B. Read Book
- C. Read E-Book
- D. Compilation fails at line n1.
- E. Compilation fails at line n3.

**Answer:** B

#### NEW QUESTION 141

Which two class definitions fail to compile?

- A. abstract class A3 {private static int i;public void doStuff(){}public A3(){}}
- B. final class A1 {public A1(){}}
- C. public class A2 {private static int i;private A2(){}}
- D. class A4 {protected static final int i;private void doStuff(){}}
- E. final abstract class A5 {protected static int i;void doStuff(){}abstract void doIt();}

**Answer:** CE

#### NEW QUESTION 145

Given:

```
public class App {
    public static void main(String[] args) {
        int i = 10;
        int j = 20;
        int k = j += i / 5;
        System.out.print(i + " : " + j + " : " + k);
    }
}
```

What is the result?

- A. 10 : 30 : 6
- B. 10 : 22 : 22
- C. 10 : 22 : 20
- D. 10 : 22 : 6

**Answer:** B

**Explanation:** Your Code ...

```
1 public class App {  
2     public static void main (String[] args) {  
3         int i = 10;  
4         int j = 20;  
5         int k = j += i / 5;  
6         System.out.print (i + " : " + j + " : " + k);  
7     }  
8 }  
9
```

External Libraries ...

CommandLine Arguments ...

Interactive mode :  OFF

Version:

JDK 9.0.1

Stdin Inputs...

Result...

CPU Time: 0.20 sec(s), Memory: 32080 kilobyte(s)

compiled and executed in 1.229 sec(s)

10 : 22 : 22

#### NEW QUESTION 148

Given the code fragments:

Person.java:

```
public class Person {  
    String name;  
    int age;  
  
    public Person(String n, int a) {  
        name = n;  
        age = a;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getAge() {  
        return age;  
    }  
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {  
    for (Person p : list) {  
        if (predicate.test(p)) {  
            System.out.println(p.name + " ");  
        }  
    }  
}  
  
public static void main(String[] args) {  
    List<Person> iList = Arrays.asList(new Person("Hank", 45),  
   new Person("Charlie", 40),  
   new Person("Smith", 38));  
    //line n1  
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

- A. checkAge (iList, () ->
- B. get Age () > 40);
- C. checkAge(iList, Person p -> p.getAge() > 40);
- D. checkAge (iList, p -> p.getAge () > 40);
- E. checkAge(iList, (Person p) -> { p.getAge() > 40; });

**Answer: C**

#### NEW QUESTION 151

fragment:

```
1. class X {  
2.     public void printFileContent() {  
3.         /* code goes here */  
4.         throw new IOException();  
5.     }  
6. }  
7. public class Test {  
8.     public static void main(String[] args) {  
9.         X xobj = new X();  
10.        xobj.printFileContent();  
11.    }  
12. }
```

Which two modifications should you make so that the code compiles successfully?

- A) Replace line 8 with `public static void main(String[] args) throws Exception {`
- B) Replace line 10 with:  

```
try {  
    xobj.printFileContent();  
}  
catch(Exception e) {}  
catch(IOException e) {}
```
- C) Replace line 2 with `public void printFileContent() throws IOException {`
- D) Replace line 4 with `throw IOException("Exception raised");`
- E) At line 11, insert `throw new IOException();`

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** AC

**NEW QUESTION 154**

Given:

```
class Vehicle {  
    String type = "4W";  
    int maxSpeed = 100;  
  
    Vehicle(String type, int maxSpeed) {  
        this.type = type;  
        this.maxSpeed = maxSpeed;  
    }  
}  
  
class Car extends Vehicle {  
    String trans;  
  
    Car(String trans) { //line n1  
        this.trans = trans;  
    }  
  
    Car(String type, int maxSpeed, String trans) {  
        super(type, maxSpeed);  
        this(trans); //line n2  
    }  
}
```

And given the code fragment:

7. Car c1 = new Car("Auto");
8. Car c2 = new Car("4W", 150, "Manual");
9. System.out.println(c1.type + " " + c1.maxSpeed + " " + c1.trans);
10. System.out.println(c2.type + " " + c2.maxSpeed + " " + c2.trans);

What is the result?

- A. 4W 100 Auto4W 150 Manual
- B. Null 0 Auto4W 150 Manual
- C. Compilation fails only at line n1
- D. Compilation fails only at line n2
- E. Compilation fails at both line n1 and line n2

**Answer:** C

#### NEW QUESTION 157

Given the code fragments:

A.java:

```
package p1;
public class A { }
```

B.java:

```
package p1.p2;
//line n1
public class B {
    public void doStuff() {
        A b = new A();
    }
}
```

C.java:

```
package p3;
//line n2
public class C {
    public static void main(String[] args) {
        A o1 = new A();
        B o2 = new B();
    }
}
```

Which modification enables the code to compile?

- A. Replace line n1 with: import p1.A;Replace line n2 with: import p1.A;import p1.p2.B;
- B. Replace line n1 with: import p1;Replace line n2 with: import p1;import p1.p2;
- C. Replace line n1 with: import p1.A;Replace line n2 with: import p1.\*;
- D. Replace line n1 with: import p1.\*;Replace line n2 with: import p1.p2.\*;

Answer: D

#### NEW QUESTION 161

Given:

```
class Animal {  
    String type = "Canine";  
    int maxSpeed = 60;  
  
    Animal () {}  
  
    Animal (String type, int maxSpeed) {  
        this.type = type;  
        this.maxSpeed = maxSpeed;  
    }  
}  
  
class WildAnimal extends Animal {  
    String bounds;  
  
    WildAnimal (String bounds) {  
        //line n1  
    }  
    WildAnimal (String type, int maxSpeed,  
        //line n2  
    }  
}
```

And given the code fragment:

7. WildAnimal wolf = new WildAnimal ("Long");
8. WildAnimal tiger = new WildAnimal ("Feline", 80, "Short");
9. System.out.println (wolf.type + " " + wolf.maxSpeed + " " + wolf.bounds);
10. System.out.println (tiger.type + " " + tiger.maxSpeed + " " + tiger.bounds);

Which two modifications enable the code to print the following output? Canine 60 Long  
Feline 80 Short

- A. Replace line n1 with:super ();this.bounds = bounds;
- B. Replace line n1 with:this.bounds = bounds;super ();
- C. Replace line n2 with:super (type, maxSpeed);this (bounds);
- D. Replace line n1 with:this ("Canine", 60);this.bounds = bounds;
- E. Replace line n2 with:super (type, maxSpeed);this.bounds = bounds;

**Answer:** A

#### NEW QUESTION 163

The following grid shows the state of a 2D array:

|   |   |   |
|---|---|---|
| 0 | 0 |   |
| X |   | 0 |
| X |   | X |

This grid is created with the following code:

```
char[][] grid = new char[3][3];
grid[1][1] = 'X';
grid[0][0] = '0';
grid[2][1] = 'X';
grid[0][1] = '0';
grid[2][2] = 'X';
grid[1][2] = '0';
```

Which line of code, when inserted in place of //line n1, adds an X into the grid so that the grid contains three consecutive X's?

- A. grid[1][3] = 'X';
- B. grid[3][1] = 'X';
- C. grid[0][2] = 'X';
- D. grid[2][0] = 'X';
- E. grid[1][2] = 'X';

**Answer:** C

#### NEW QUESTION 167

Given the following main method:

```
public static void main(String[] args) {
    int num = 5;
    do {
        System.out.print(num-- + " ");
    } while (num == 0);
}
```

What is the result?

- A. 5 4 3 2 1 0
- B. 5 4 3 2 1
- C. 4 2 1
- D. 5
- E. Nothing is printed

**Answer:** D

#### NEW QUESTION 170

Given the code fragment:

```
24. float var1 = (12_345.01 >= 123_45.00) ? 12_456 : 124_56.02f;
25. float var2 = var1 + 1024;
26. System.out.print(var2);
```

What is the result?

- A. An exception is thrown at runtime.
- B. Compilation fail
- C. 13480.0
- D. 13480.02

**Answer:** C

#### NEW QUESTION 173

Given:

```
public class TestScope {  
    public static void main(String[] args) {  
        int var1 = 200;  
        System.out.print(doCalc(var1));  
        System.out.print(" "+var1);  
    }  
    static int doCalc(int var1){  
        var1 = var1 * 2;  
        return var1;  
    }  
}
```

What is the result?

- A. 400 200
- B. 200 200
- C. 400 400
- D. Compilation fails.

**Answer:** A

#### NEW QUESTION 176

Given the code fragment:

```
abstract class Toy {  
    int price;  
    // line n1  
}
```

Which three code fragments are valid at line n1?

- A. public static void insertToy() /\* code goes here \*/
- B. public abstract Toy getToy() {return new Toy();}
- C. public void printToy();
- D. public int calculatePrice() {return price;}
- E. public abstract int computeDiscount();

**Answer:** CDE

#### NEW QUESTION 178

Given:

```
class A {  
    public void test () {  
        System.out.println ("A");  
    }  
}  
class B extends A {  
    public void test () {  
        System.out.println ("B");  
    }  
}  
public class C extends A {  
    public void test () {  
        System.out.println ("C");  
    }  
  
    public static void main (String [] args) {  
        A b1 = new A ();  
        A b2 = new C ();  
        b1 = (A) b2;  
        A b3 = (B) b2; //line n1  
        A b3 = (B) b2; //line n2  
        b1.test ();  
        b3.test ();  
    }  
}
```

What is the result?

- A. AB
- B. AC
- C. CC
- D. A ClassCastException is thrown only at line n1.
- E. A ClassCastException is thrown only at line n2.

**Answer:** E

#### NEW QUESTION 180

Given:

```
package clothing;  
public class Shirt {  
    public static String getColor() {  
        return "Green";  
    }  
}
```

Given the code fragment:

```
package clothing.pants;
// line n1
public class Jeans {
    public void matchShirt() {
        //line n2
        if(color.equals("Green")) {
            System.out.print("Fit")
        }
    }
    public static void main (String[] args) {
        Jeans trouser = new Jeans();
        trouser.matchShirt();
    }
}
```

Which two sets of actions, independently, enable the code fragment to print Fit?

- A. At line n1 insert:import clothing.Shirt;At line n2 insert:String color = getColor();
- B. At line n1 insert:import clothing.\*;At line n2 insert:String color = Shirt.getColor();
- C. At line n1 insert:import static clothing.Shirt.getColor();At line n2 insert:String color = getColor();
- D. At line n1 no changes required.At line n2 insert:String color = Shirt.getColor();
- E. At line n1 insert:import clothing;At line n2 insert:String color = Shirt.getColor();

**Answer:** A

#### NEW QUESTION 181

Given:

```
class CD {
    int r;
    CD(int r) {
        this.r=r;
    }
}
```

```
class DVD extends CD {
    int c;
    DVD(int r, int c) {
        // line n1
    }
}
```

And given the code fragment:

```
DVD dvd = new DVD(10, 20);
```

Which code fragment should you use at line n1 to instantiate the dvd object successfully?

- A) super.r = r;  
this.c = c;
- B) super(r);  
this(c);
- C) super(r);  
this.c = c;
- D) this.c = r;  
super(c);

- A. Option A  
B. Option B  
C. Option C  
D. Option D

Answer: C

#### NEW QUESTION 186

Given:

```
public class Test {  
    public static void main(String[] args) {  
        boolean a = new Boolean(Boolean.valueOf(args[0]));  
        boolean b = new Boolean(args[1]);  
        System.out.println(a + " " + b);  
    }  
}
```

And given the commands: javac Test.java  
java Test TRUE null What is the result?

- A. TRUE null  
B. true false  
C. false false  
D. true true  
E. AClassCastExceptionis thrown at runtime.

Answer: D

#### NEW QUESTION 190

Given:

```
public class Test {  
  
    public static void main(String[] args) {  
  
        String[][] chs = new String[2][];  
        chs[0] = new String[2];  
        chs[1] = new String[5];  
        int i = 97;  
  
        for (int a = 0; a < chs.length; a++) {  
            for (int b = 0; b < chs.length; b++) {  
                chs[a][b] = "" + i;  
                i++;  
            }  
        }  
  
        for (String[] ca : chs) {  
            for (String c : ca) {  
                System.out.print(c + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

What is the result?

- A. 97 98 99 100 null null null
- B. 97 98 99 100 101 102 103
- C. Compilation fails.
- D. A NullPointerException is thrown at runtime.
- E. An ArrayIndexOutOfBoundsException is thrown at runtime.

**Answer:** A

#### NEW QUESTION 191

Which statement is true about the switch statement?

- A. It must contain the default section.
- B. The break statement, at the end of each case block, is mandatory.
- C. Its case label literals can be changed at runtime.
- D. Its expression must evaluate to a single value.

**Answer:** D

#### NEW QUESTION 193

Given the code fragment:

```
public static void main (String [] args) {  
    ArrayList<Integer> points = new ArrayList<> ();  
    points.add (1);  
    points.add (2);  
    points.add (3);  
    points.add (4);  
    points.add (null);  
    points.remove (2);  
    points.remove (null);  
    System.out.println(points);  
}
```

What is the result?

- A. A NullPointerException is thrown at runtime.
- B. [1, 2, 4]
- C. [1, 2, 4, null ]
- D. [1, 3, 4, null ]
- E. [1, 3, 4 ]
- F. Compilation fails.

Answer: F

Explanation:

Version - JDK 1.8.0\_66

Your Code ...

```
1 - public static void main (String [] args) {  
2     ArrayList<Integer> points = new ArrayList<> ();  
3     points.add (1) ;  
4     points.add (2) ;  
5     points.add (3) ;  
6     points.add (4) ;  
7     points.add (null) ;  
8     points.remove (null) ;  
9     System.out.println (points) ;  
10 }
```

External Libraries ...

Add External Library (from Maven Repo)

cs1.keyboard

Input Arguments (args of Main Method)...

Interactive mode :  OFF

Stdin Inputs...

**Execute**

**Save**

**My Projects**

**Recent**

**Collaborate**

**Others** ▾

**Goto Another Language/DB ▾**

Result...

compiled and executed in @ second(s)

No "public class" found to execute

#### NEW QUESTION 195

Which code fragment causes a compilation error?

- A. float flt = 100F;
- B. float flt = (float) 1\_11.00;
- C. float flt = 100;
- D. double y1 = 203.22;  
    float flt = y1;
- E. int y2 = 100;  
    float flt = (float) y2;

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** D

#### NEW QUESTION 198

Given the code fragment:

```
String[] strs = new String[2];
int idx = 0;
for (String s : strs) {
    strs[idx].concat(" element " + idx);
    idx++;
}
for (idx = 0; idx < strs.length; idx++) {
    System.out.println(strs[idx]);
}
```

What is the result?

- A. Element 0Element 1
- B. Null element 0Null element 1
- C. NullNull
- D. A NullPointerException is thrown at runtime.

**Answer:** C

#### NEW QUESTION 200

Given the code fragment:

```
int a[] = {1, 2, 3, 4, 5};
for(XXX) {
    System.out.print(a[e]);
}
```

Which option can replace xxx to enable the code to print 135?

- A. int e = 0; e <= 4; e++
- B. int e = 0; e < 5; e += 2
- C. int e = 1; e <= 5; e += 1
- D. int e = 1; e < 5; e+ =2

**Answer:** B

#### NEW QUESTION 205

What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?

- A. Encapsulation

- B. Inheritance
- C. Abstraction
- D. Instantiation
- E. Polymorphism

**Answer:** A

**Explanation:** Explanation

Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

**NEW QUESTION 207**

Given the following class declarations: Which answer fails to compile?

- A) `ArrayList<Animal> myList = new ArrayList<>();  
myList.add(new Tiger());`
- B) `ArrayList<Hunter> myList = new ArrayList<>();  
myList.add(new Cat());`
- C) `ArrayList<Hunter> myList = new ArrayList<>();  
myList.add(new Tiger());`
- D) `ArrayList<Tiger> myList = new ArrayList<>();  
myList.add(new Cat());`
- E) `ArrayList<Animal> myList = new ArrayList<>();  
myList.add(new Cat());`

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** E

**NEW QUESTION 211**

Given:

```
class Product {  
    double price;  
}  
  
public class Test {  
    public void updatePrice(Product product, double price) {  
        price = price * 2;  
        product.price = product.price + price;  
    }  
    public static void main(String[] args) {  
        Product prt = new Product();  
        prt.price = 200;  
        double newPrice = 100;  
  
        Test t = new Test();  
        t.updatePrice(prt, newPrice);  
        System.out.println(prt.price + " : " + newPrice);  
    }  
}
```

What is the result?

- A. 200.0 : 100.0
- B. 400.0 : 200.0
- C. 400.0 : 100.0
- D. Compilation fails.

Answer: C

#### NEW QUESTION 215

Given:

```
public class Test {  
    public static void main(String[] args) {  
        int x = 1;  
        int y = 0;  
        if(x++ > ++y) {  
            System.out.print("Hello ");  
        } else {  
            System.out.print("Welcome ");  
        }  
        System.out.print("Log " + x + ":" + y);  
    }  
}
```

What is the result?

- A. Hello Log 1:0
- B. Hello Log 2:1
- C. Welcome Log 2:1
- D. Welcome Log 1:0

Answer: C

#### NEW QUESTION 218

Given the code fragment:

```
public class Test {  
  
    static int count = 0  
    int i = 0;  
  
    public void changeCount () {  
        while (i<5) {  
            i++;  
            count++;  
        }  
    }  
  
    public static void main (String [] args) {  
        Test check1 = new Test ();  
        Test check2 = new Test ();  
        check1.changeCount ();  
        check2.changeCount ();  
        System.out. print (check1.count + " : " + check2.count);  
    }  
}
```

What is the result?

- A. 5 : 5
- B. 10 : 10
- C. 5 : 10
- D. Compilation fails.

**Answer:** B

#### NEW QUESTION 220

Given the content of three files:

A.java:

```
public class A {  
    public void a() {}  
    int a;  
}
```

B.java:

```
public class B {  
    private int doStuff() {  
        private int x = 100;  
        return x++;  
    }  
}
```

C.java:

```
import java.io.*;  
package p1;  
class A {  
    public void main(String fileName) throws IOException {}  
}
```

Which statement is true?

- A. Only the A.java file compiles successfully.
- B. Only the B.java file compiles successfully.
- C. Only the C.java file compiles successfully.
- D. The A.java and B.java files compile successfully.
- E. The B.java and C.java files compile successfully.
- F. The A.java and C.java files compile successfully.

Answer: A

#### NEW QUESTION 225

Given:

```
public class Triangle {  
    static double area;  
    int b = 2, h = 3;  
    public static void main(String[] args) {  
        double p, b, h; //line n1  
        if (area == 0) {  
            b = 3;  
            h = 4;  
            p = 0.5;  
        }  
        area = p * b * h; //line n2  
        System.out.println("Area is " + area);  
    }  
}
```

What is the result?

- A. Area is 6.0
- B. Area is 3.0
- C. Compilation fails at line n1
- D. Compilation fails at line n2.

Answer: D

**NEW QUESTION 230**

Given the code fragment:

```
int wd = 0;
String days[] = {"sun", "mon", "wed", "sat"};
for (String s:days) {
    switch (s) {
        case "sat":
        case "sun":
            wd -= 1;
            break;
        case "mon":
            wd++;
        case "wed":
            wd += 2;
    }
}
System.out.println(wd);
```

What is the result?

- A. 3
- B. 4
- C. -1
- D. Compilation fails.

**Answer:** B**NEW QUESTION 235**

Given the code fragment:

```
LocalDate date1 = LocalDate.now();
LocalDate date2 = LocalDate.of(2014, 6, 20);
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);
System.out.println("date1 = " + date1);
System.out.println("date2 = " + date2);
System.out.println("date3 = " + date3);
```

Assume that the system date is June 20, 2014. What is the result?

- A) date1 = 2014-06-20  
date2 = 2014-06-20  
date3 = 2014-06-20
- B) date1 = 06/20/2014  
date2 = 2014-06-20  
date3 = Jun 20, 2014
- C) Compilation fails.
- D) A DateParseException is thrown at runtime.

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** D**NEW QUESTION 240**

.....

## Thank You for Trying Our Product

We offer two products:

1st - We have Practice Tests Software with Actual Exam Questions

2nd - Questions and Answers in PDF Format

### 1Z0-808 Practice Exam Features:

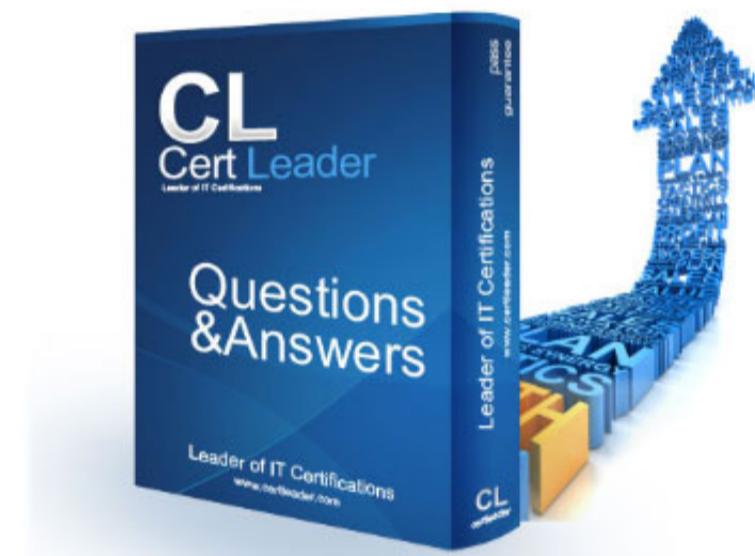
- \* 1Z0-808 Questions and Answers Updated Frequently
- \* 1Z0-808 Practice Questions Verified by Expert Senior Certified Staff
- \* 1Z0-808 Most Realistic Questions that Guarantee you a Pass on Your First Try
- \* 1Z0-808 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

100% Actual & Verified — Instant Download, Please Click  
[Order The 1Z0-808 Practice Test Here](#)

## 1z0-808 Dumps

### Java SE 8 Programmer I

<https://www.certleader.com/1z0-808-dumps.html>



**NEW QUESTION 1**

Which one of the following code examples uses valid Java syntax?

- A.
- ```
public class Boat {  
  
    public static void main (String [] args) {  
        System.out.println ("I float.");  
    }  
}
```
- B.
- ```
public class Cake {  
    public static void main (String [] ) {  
        System.out.println ("Chocolate");  
    }  
}
```
- C.
- ```
public class Dog {  
    public void main (String [] args) {  
        System.out.println ("Squirrel.");  
    }  
}
```
- D.
- ```
public class Bank {  
    public static void main (String () args) {  
        System.out.println ("Earn interest.");  
    }  
}
```

- A. Option A  
B. Option B  
C. Option C  
D. Option D

**Answer:** A

**NEW QUESTION 2**

Given:

```
public static void main(String[] args) {  
    String ta = "A ";  
    ta = ta.concat("B ");  
    String tb = "C ";  
    ta = ta.concat(tb);  
    ta.replace('C', 'D');  
    ta = ta.concat(tb);  
    System.out.println(ta);  
}
```

What is the result?

- A. A B C D  
B. A C D  
C. A C D D  
D. A B D  
E. A B D C

**Answer:** C

**NEW QUESTION 3**

You are asked to create a method that accepts an array of integers and returns the highest value from that array.  
Given the code fragment:

```
class Test{
    public static void main(String[] args) {
        int numbers[] = {12, 13, 42, 32, 15, 156, 23, 51, 12};
        int[] keys = findMax(numbers);
    }

    /* line n1 */
    int[] keys = new int[3];
    /* code goes here*/
    return keys;
}
}
```

Which method signature do you use at line n1?

- A. public int findMax (int[] numbers)
- B. static int[] findMax (int[] max)
- C. static int findMax (int[] numbers)
- D. final int findMax (int[] )

**Answer:** C

#### NEW QUESTION 4

Given the code fragments:

```
class Student {
    String name;
    int age;
}
```

And:

```
4. public class Test {
5.     public static void main(String[] args) {
6.         Student s1 = new Student();
7.         Student s2 = new Student();
8.         Student s3 = new Student();
9.         s1 = s3;
10.        s3 = s2;
11.        s2 = null;
12.    }
13.}
```

Which statement is true?

- A. After line 11, three objects are eligible for garbage collection.
- B. After line 11, two objects are eligible for garbage collection.
- C. After line 11, one object is eligible for garbage collection.
- D. After line 11, none of the objects are eligible for garbage collection.

**Answer:** C

#### NEW QUESTION 5

You are asked to develop a program for a shopping application, and you are given this information:

- The application must contain the classes Toy, EduToy, and ConsToy. The Toy class is the superclass of the other two classes.
- The int calculatePrice (Toy t) method calculates the price of a toy.
- The void printToy (Toy t) method prints the details of a toy.

Which definition of the Toy class adds a valid layer of abstraction to the class hierarchy?

A

```
public abstract class Toy{  
    public abstract int calculatePrice(Toy t);  
    public void printToy(Toy t) { /* code goes here */ }  
}
```

B

```
public abstract class Toy {  
    public int calculatePrice(Toy t) ;  
    public void printToy(Toy t) ;  
}
```

C

```
public abstract class Toy {  
    public int calculatePrice(Toy t);  
    public final void printToy(Toy t){ /* code goes here */ }  
}
```

D

```
public abstract class Toy {  
    public abstract int calculatePrice(Toy t) { /* code goes here */ }  
    public abstract void printToy(Toy t) { /* code goes here */ }  
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A**NEW QUESTION 6**

Given:

```
String stuff = "TV";  
String res = null;  
  
if (stuff.equals("TV")) {  
    res = "Walter";  
} else if (stuff.equals("Movie")) {  
    res = "White";  
} else {  
    res = "No Result";  
}
```

Which code fragment can replace the if block?

A

```
stuff.equals ("TV") ? res= "Walter" : stuff.equals ("Movie") ?  
res = "White" : res = "No Result";
```

B

```
res = stuff.equals ("TV") ? "Walter" else stuff.equals  
("Movie")? "White" : "No Result";
```

C

```
res = stuff.equals ("TV") ? stuff.equals ("Movie")? "Walter" :  
"White" : "No Result";
```

D

```
res = stuff.equals ("TV")? "Walter" : stuff.equals ("Movie")?  
"White" : "No Result";
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** D**NEW QUESTION 7**

Given:

```
public class Test {  
    public static void main(String[] args) {  
        int x = 1;  
        int y = 0;  
        if(x++ > ++y) {  
            System.out.print("Hello ");  
        } else {  
            System.out.print("Welcome ");  
        }  
        System.out.print("Log " + x + ":" + y);  
    }  
}
```

What is the result?

- A. Hello Log 1:0
- B. Hello Log 2:1
- C. Welcome Log 2:1
- D. Welcome Log 1:0

**Answer:** C

#### NEW QUESTION 8

Given the code fragment:

```
LocalDate Time dt= LocalDateTime.of (2014, 7, 31, 1, 1);  
dt.plusDays (30);  
dt. plusMonths (1);  
System.out.print (dt format (DateTimeFormatter. ISO_DATE) );
```

What is the result?

- A. An exception is thrown at runtime
- B. 07-31-2014
- C. 2014-07-31
- D. 2014-09-30

**Answer:** A

#### NEW QUESTION 9

Given the code fragment:

```
public static void main(String[] args) {  
    Short s1 = 200;  
    Integer s2 = 400;  
    Long s3 = (long) s1 + s2;           //line n1  
    String s4 = (String) (s3 * s2);     //line n2  
    System.out.println("Sum is " + s4);  
}
```

What is the result?

- A. Sum is 600
- B. Compilation fails at line n1.
- C. Compilation fails at line n2.
- D. A ClassCastException is thrown at line n1.
- E. A ClassCastException is thrown at line n2.

**Answer:** C

#### NEW QUESTION 10

Given:

```
public class App {  
    int count;  
    public static void displayMsg () {  
        count++; // line n1  
        System.out.println ("Welcome "+"Visit Count: "+count); // line n2  
    }  
    public static void main (String [] args) {  
        App.displayMsg (); // line n3  
        App.displayMsg (); // line n4  
    }  
}
```

What is the result?

- A. Compilation fails at line n3 and line n4.
- B. Compilation fails at line n1 and line n2.
- C. Welcome Visit Count:1Welcome Visit Count: 1
- D. Welcome Visit Count:1Welcome Visit Count: 2

**Answer:** B

**NEW QUESTION 10**

Which two class definitions fail to compile? (Choose two.)

A

```
abstract class A3 {  
    private static int i;  
    public void doStuff() {}  
    public A3() {}  
}
```

B

```
final class A1 {  
    public A1() {}  
}
```

C

```
private class A2 {  
    private static int i;  
    private A2() {}  
}
```

D

```
class A4 {  
    protected static final int i = 10;  
    private A4() {}  
}
```

E

```
final abstract class A5 {  
    protected static int i;  
    void doStuff() {}  
    abstract void doIt();  
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** CD

**NEW QUESTION 12**

Given the code fragment:

```
public static void main(String[] args) {
    int ii = 0;
    int jj = 7;
    for (ii = 0; ii < jj - 1; ii = ii + 2) {
        System.out.print(ii + " ");
    }
}
```

What is the result?

- A. 2 4
- B. 0 2 4 6
- C. 0 2 4
- D. Compilation fails

**Answer:** C

#### NEW QUESTION 16

Given:

```
public class App {
    int count;
    public static void displayMsg() {
        System.out.println("Welcome Visit Count: " + count++); // line n1
    }
    public static void main(String[] args) {
        App.displayMsg();
        displayMsg(); // line n2
    }
}
```

What is the result?

- A. Welcome Visit Count:0Welcome Visit Count: 1
- B. Compilation fails at line n2.
- C. Compilation fails at line n1.
- D. Welcome Visit Count:0Welcome Visit Count: 0

**Answer:** C

#### Explanation:

```
1
2 public class App {
3     int count;
4     public static void displayMsg() {
5         System.out.println("Welcome Visit Count: " + count ++); //line n1
6     }
7     public static void main(String[] args) {
8         App.displayMsg();
9         displayMsg();
10    }
11 }
12
```

#### NEW QUESTION 19

Given the code fragment:

```
public static void main(String[] args) {
    LocalDate date = LocalDate.of(2012, 1, 30);
    date.plusDays(10);
    System.out.println(date);
}
```

What is the result?

- A. 2012-02-10 00:00
- B. 2012-01-30
- C. 2012-02-10
- D. A DateTimeException is thrown at runtime.

**Answer:** B

#### Explanation:



Main.java

```
1 import java.time.LocalDate;
2 import java.time.Month;
3
4 public class Main {
5     public static void main(String[] args) {
6         LocalDate date = LocalDate.of(2012, 1, 30);
7         date.plusDays(10);
8         System.out.println(date);
9     }
10 }
```

java version "1.8.0\_31"
Java(TM) SE Runtime Environment (build 1.8.0\_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)
> javac -classpath .:/run\_dir/junit-4.12.jar:/run\_dir/hamcrest-core-1.3.jar:/run\_dir/json-simple-1.1.1.jar -d . Main.java
> java -classpath .:/run\_dir/junit-4.12.jar:/run\_dir/hamcrest-core-1.3.jar:/run\_dir/json-simple-1.1.1.jar Main
2012-01-30

**NEW QUESTION 21**

Given:

```
public class MyClass {
    public static void main(String[] args) {
        String s = "Java SE 8 1";
        int len = s.trim().length();
        System.out.print(len);
    }
}
```

What is the result?

- A. Compilation fails.
- B. 11
- C. 8
- D. 9
- E. 10

**Answer: B****NEW QUESTION 22**

Given the code fragment:

```
public class Employee {
    String name;
    boolean contract;
    double salary;
    Employee() {
        // line n1
    }
    public String toString() {
        return name + ":" + contract + ":" + salary;
    }
    public static void main(String[] args) {
        Employee e = new Employee();
        // line n2
        System.out.print(e);
    }
}
```

Which two modifications, when made independently, enable the code to print Joe:true: 100.0? (Choose two.)

- A) Replace line n2 with:  
e.name = "Joe";  
e.contract = true;  
e.salary = 100;
- B) Replace line n2 with:  
this.name = "Joe";  
this.contract = true;  
this.salary = 100;
- C) Replace line n1 with:  
this.name = new String("Joe");  
this.contract = new Boolean(true);  
this.salary = new Double(100);
- D) Replace line n1 with:  
name = "Joe";  
contract = TRUE;  
salary = 100.0f;
- E) Replace line n1 with:  
this("Joe", true, 100);

- A. Option A  
B. Option B  
C. Option C  
D. Option D  
E. Option E

**Answer:** AC

#### NEW QUESTION 23

Given:

```
class X {  
    static int i;  
    int j;  
    public static void main(String[] args) {  
        X x1 = new X();  
        X x2 = new X();  
        x1.i = 3;  
        x1.j = 4;  
        x2.i = 5;  
        x2.j = 6;  
        System.out.println(  
            x1.i + " " +  
            x1.j + " " +  
            x2.i + " " +  
            x2.j);  
    }  
}
```

What is the result?

- A. 3 4 5 6  
B. 3 4 3 6  
C. 5 4 5 6  
D. 3 6 4 6

**Answer:** C

#### NEW QUESTION 28

Which three statements are true about exception handling? (Choose three.)

- A. Only unchecked exceptions can be rethrown.  
B. All subclasses of the RuntimeException class are not recoverable.  
C. The parameter in a catch block is of Throwable type.  
D. All subclasses of the RuntimeException class must be caught or declared to be thrown.  
E. All subclasses of the RuntimeException class are unchecked exceptions.  
F. All subclasses of the Error class are not recoverable.

**Answer:** BCD

#### NEW QUESTION 31

Given:

```
class A {
    public void test() {
        System.out.println("A ");
    }
}

class B extends A {
    public void test() {
        System.out.println("B ");
    }
}

public class C extends A {
    public void test() {
        System.out.println("C ");
    }
}

public static void main(String[] args) {
    A b1 = new A();
    A b2 = new C();
    A b3 = (B) b2;           //line n1
    b1 = (A) b2;           //line n2
    b1.test();
    b3.test();
}
```

What is the result?

- A. AB
- B. AC
- C. CC
- D. A ClassCastException is thrown only at line n1.
- E. A ClassCastException is thrown only at line n2.

**Answer:** D

#### NEW QUESTION 35

Given the code fragment:

```
abstract class Toy {
    int price;
    // line n1
}
```

Which three code fragments are valid at line n1?

A

```
public static void insertToy() {
    /* code goes here */
}
```

B

```
final Toy getToy() {
    return new Toy();
}
```

C

```
public void printToy();
```

D

```
public int calculatePrice() {
    return price;
}
```

E

```
public abstract int computeDiscount();
```

A. Option A

- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** CDE

#### NEW QUESTION 37

Given:

```
class Caller {  
    private void init () {  
        System.out.println("Initialized");  
    }  
  
    private void start () {  
        init();  
        System.out.println("Started");  
    }  
}  
  
public class TestCall {  
    public static void main(String[] args) {  
        Caller c = new Caller();  
        c.start(); // line n1  
        c.init(); // line n2  
    }  
}
```

What is the result?

- A. Compilation fails at line n1.
- B. InitializedStartedInitialized
- C. InitializedStarted
- D. Compilation fails at line n2.

**Answer:** D

#### NEW QUESTION 40

Given this class:

```
public class CheckingAccount {  
    public int amount;  
    //line n1  
}
```

And given this main method, located in another class:

```
public static void main(String[] args) {  
    CheckingAccount acct = new CheckingAccount();  
    //line n2  
}
```

Which three pieces of code, when inserted independently, set the value of amount to 100?

A

At line n1 insert:  
public CheckingAccount() {  
 amount = 100;  
}

B

At line n2 insert:  
this.amount = 100;

C

At line n2 insert:  
amount = 100;

D

At line n1 insert:  
public CheckingAccount() {  
 this.amount = 100;  
}

E

At line n2 insert:  
acct.amount = 100;

F

At line n1 insert:  
public CheckingAccount() {  
 acct.amount = 100;  
}

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

**Answer:** DE**NEW QUESTION 42**

Given:

```
class Test {  
    int a1;  
  
    public static void doProduct(int a) {  
        a = a * a;  
    }  
  
    public static void doString(String s) {  
        s.concat(" " + s);  
    }  
  
    public static void main(String[] args) {  
        Test item = new Test();  
        item.a1 = 11;  
        String sb = "Hello";  
        Integer i = 10;  
        doProduct(i);  
        doString(sb);  
        doProduct(item.a1);  
        System.out.println(i + " " + sb + " " + item.a1);  
    }  
}
```

What is the result?

- A. 10 Hello Hello 11
- B. 10 Hello Hello 121
- C. 100 Hello 121
- D. 100 Hello Hello 121
- E. 10 Hello 11

**Answer:** E

#### NEW QUESTION 46

Given:

```
public class Triangle {  
    static double area;  
    int b = 2, h = 3;  
    public static void main(String[] args) {  
        double p, b, h; //line n1  
        if (area == 0) {  
            b = 3;  
            h = 4;  
            p = 0.5;  
            area = p * b * h; //line n2  
        }  
        System.out.println("Area is " + area);  
    }  
}
```

What is the result?

- A. Area is 6.0
- B. Area is 3.0
- C. Compilation fails at line n1
- D. Compilation fails at line n2.

**Answer:** D

#### NEW QUESTION 48

Given the code fragment:

```
public static void main(String[] args) {  
    String myStr = "Hello World ";  
    myStr.trim();  
    int i1 = myStr.indexOf(" ");  
    System.out.println(i1);  
}
```

What is the result?

- A. An exception is thrown at runtime.
- B. -1
- C. 5
- D. 10

**Answer:** A

#### NEW QUESTION 51

What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?

- A. Encapsulation
- B. Inheritance
- C. Abstraction
- D. Instantiation
- E. Polymorphism

**Answer:** A

#### Explanation:

Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

#### NEW QUESTION 54

Given the code fragment:

```
int wd = 0;
String days[] = {"sun", "mon", "wed", "sat"};
for (String s:days) {
    switch (s) {
        case "sat":
        case "sun":
            wd -= 1;
            break;
        case "mon":
            wd++;
        case "wed":
            wd += 2;
    }
}
System.out.println(wd);
```

What is the result?

- A. 3
- B. 4
- C. -1
- D. Compilation fails.

**Answer:** A

#### NEW QUESTION 59

Given:

```
public class Test {
    int x, y;

    public Test(int x, int y) {
        initialize(x, y);
    }

    public void initialize(int x, int y) {
        this.x = x * x;
        this.y = y * y;
    }

    public static void main(String[] args) {
        int x = 3, y = 5;
        Test obj = new Test(x, y);
        System.out.println(x + " " + y);
    }
}
```

What is the result?

- A. Compilation fails.
- B. 3 5
- C. 0 0
- D. 9 25

**Answer:** B

#### NEW QUESTION 60

Given the code fragment:

```
public static void main(String[] args) {
    StringBuilder sb = new StringBuilder("Java");
    String s = "Java";

    if (sb.toString().equals(s.toString())) {
        System.out.println("Match 1");
    } else if (sb.equals(s)) {
        System.out.println("Match 2");
    } else {
        System.out.println("No Match");
    }
}
```

What is the result?

- A. Match 1
- B. Match 2

- C. No Match  
D. A NullPointerException is thrown at runtime.

**Answer:** A

#### NEW QUESTION 61

Given this class:

```
public class Rectangle {  
    private double length;  
    private double height;  
    private double area;  
  
    public void setLength(double length) {  
        this.length = length;  
    }  
    public void setHeight(double height) {  
        this.height = height;  
    }  
    public void setArea() {  
        area = length*height;  
    }  
}
```

Which two changes would encapsulate this class and ensure that the area field is always equal to length \* height whenever the Rectangle class is used?

- A. Call the setArea method at the end of the setHeight method.  
B. Call the setArea method at the beginning of the setHeight method.  
C. Call the setArea method at the end of the setLength method.  
D. Call the setArea method at the beginning of the setLength method.  
E. Change the setArea method to private.  
F. Change the area field to public.

**Answer:** AE

#### NEW QUESTION 63

Which statement best describes encapsulation?

- A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects.  
B. Encapsulation ensures that classes can be designed so that their methods are inheritable.  
C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.  
D. Encapsulation ensures that classes can be designed so that if a method has an argument MyType x, any subclass of MyType can be passed to that method.

**Answer:** A

#### NEW QUESTION 67

Which two statements are true? (Choose two.)

- A. Error class is unextendable.  
B. Error class is extendable.  
C. Error is a RuntimeException.  
D. Error is an Exception.  
E. Error is a Throwable.

**Answer:** BC

#### NEW QUESTION 72

Given the code fragment:

```
String[] strs = {"A", "B"};  
int idx = 0;  
for (String s : strs) {  
    strs[idx].concat(" element " + idx);  
    idx++;  
}  
for (idx = 0; idx < strs.length; idx++) {  
    System.out.println(strs[idx]);  
}
```

What is the result?

- A. AB  
B. A element 0B element 1  
C. A NullPointerException is thrown at runtime.  
D. A 0B 1

**Answer:** C

**NEW QUESTION 75**

Which three statements are true about the structure of a Java class? (Choose three.)

- A. A class cannot have the same name as its field.
- B. A public class must have a main method.
- C. A class can have final static methods.
- D. A class can have overloaded private constructors.
- E. Fields need to be initialized before use.
- F. Methods and fields are optional components of a class.

**Answer:** BDE

**NEW QUESTION 80**

.....

## Thank You for Trying Our Product

\* **100% Pass or Money Back**

All our products come with a 90-day Money Back Guarantee.

\* **One year free update**

You can enjoy free update one year. 24x7 online support.

\* **Trusted by Millions**

We currently serve more than 30,000,000 customers.

\* **Shop Securely**

All transactions are protected by VeriSign!

### 100% Pass Your 1z0-808 Exam with Our Prep Materials Via below:

<https://www.certleader.com/1z0-808-dumps.html>

# Exam Questions 1z0-808

Java SE 8 Programmer I

<https://www.2passeeasy.com/dumps/1z0-808/>



**NEW QUESTION 1**

Given:

```
public static void main(String[] args) {  
    String ta = "A ";  
    ta = ta.concat("B ");  
    String tb = "C ";  
    ta = ta.concat(tb);  
    ta.replace('C', 'D');  
    ta = ta.concat(tb);  
    System.out.println(ta);  
}
```

What is the result?

- A. A B C D
- B. A C D
- C. A C D D
- D. A B D
- E. A B D C

**Answer:** C

**NEW QUESTION 2**

Given the code fragments:

```
class Student {  
    String name;  
    int age;  
}
```

And:

```
4. public class Test {  
5.     public static void main(String[] args) {  
6.         Student s1 = new Student();  
7.         Student s2 = new Student();  
8.         Student s3 = new Student();  
9.         s1 = s3;  
10.        s3 = s2;  
11.        s2 = null;  
12.    }  
13.}
```

Which statement is true?

- A. After line 11, three objects are eligible for garbage collection.
- B. After line 11, two objects are eligible for garbage collection.
- C. After line 11, one object is eligible for garbage collection.
- D. After line 11, none of the objects are eligible for garbage collection.

**Answer:** C

**NEW QUESTION 3**

Given the following classes:

```
public class Employee {  
    public int salary;  
}  
  
public class Manager extends Employee {  
    public int budget;  
}  
  
public class Director extends Manager {  
    public int stockOptions;  
}
```

And given the following main method:

```
public static void main(String[] args) {  
    Employee employee = new Employee();  
    Manager manager = new Manager();  
    Director director = new Director();  
    //line n1  
}
```

Which two options fail to compile when placed at line n1 of the main method? (Choose two.)

- A. employee.salary = 50\_000;
- B. director.salary = 80\_000;
- C. employee.budget = 200\_000;
- D. manager.budget = 1\_000\_000;
- E. manager.stockOption = 500;
- F. director.stockOptions = 1\_000;

**Answer:** CE

#### NEW QUESTION 4

Given the following main method:

```
public static void main(String[] args) {  
    int num = 5;  
    do {  
        System.out.print(num-- + " ");  
    } while (num == 0);  
}
```

What is the result?

- A. 5 4 3 2 1 0
- B. 5 4 3 2 1
- C. 4 2 1
- D. 5
- E. Nothing is printed

**Answer:** D

#### NEW QUESTION 5

Given the code fragments:

Person.java:

```
public class Person {  
    String name;  
    int age;  
  
    public Person(String n, int a) {  
        name = n;  
        age = a;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getAge() {  
        return age;  
    }  
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {  
    for (Person p : list) {  
        if (predicate.test(p)) {  
            System.out.println(p.name + " ");  
        }  
    }  
}  
  
public static void main(String[] args) {  
    List<Person> iList = Arrays.asList(new Person("Hank", 45),  
   new Person("Charlie", 40),  
   new Person("Smith", 38));  
    //line n1  
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

- A  
checkAge (iList, ( ) -> p. get Age ( ) > 40);
- B  
checkAge(iList, Person p -> p.getAge( ) > 40);
- C  
checkAge (iList, p -> p.getAge ( ) > 40);
- D  
checkAge(iList, (Person p) -> { p.getAge() > 40; });

- A. Option A  
B. Option B  
C. Option C  
D. Option D

Answer: C

#### NEW QUESTION 6

Given:

```
public class Test {  
    public static void main(String[] args) {  
        int x = 1;  
        int y = 0;  
        if(x++ > ++y) {  
            System.out.print("Hello ");  
        } else {  
            System.out.print("Welcome ");  
        }  
        System.out.print("Log " + x + ":" + y);  
    }  
}
```

What is the result?

- A. Hello Log 1:0
- B. Hello Log 2:1
- C. Welcome Log 2:1
- D. Welcome Log 1:0

**Answer:** C

#### NEW QUESTION 7

Given the code fragment:

```
int x = 100;  
int a = x++;  
int b = ++x;  
int c = x++;  
int d = (a < b) ? (a < c) ? a: (b < c )? b: c: x;  
System.out.println(d);
```

What is the result?

- A. 100
- B. 101
- C. 102
- D. 103
- E. Compilation fails

**Answer:** E

#### NEW QUESTION 8

Given the code fragment:

```
public static void main(String[] args) {  
    Short s1 = 200;  
    Integer s2 = 400;  
    Long s3 = (long) s1 + s2;           //line n1  
    String s4 = (String) (s3 * s2);    //line n2  
    System.out.println("Sum is " + s4);  
}
```

What is the result?

- A. Sum is 600
- B. Compilation fails at line n1.
- C. Compilation fails at line n2.
- D. A ClassCastException is thrown at line n1.
- E. A ClassCastException is thrown at line n2.

**Answer:** C

#### NEW QUESTION 9

Given:

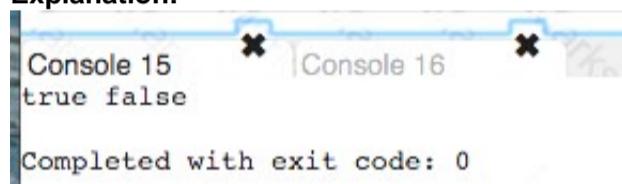
```
public class Test {  
    public static void main(String[] args) {  
        Test ts = new Test();  
        System.out.print(isAvailable + " ");  
        isAvailable= ts.doStuff();  
        System.out.println(isAvailable);  
    }  
    public static boolean doStuff() {  
        return !isAvailable;  
    }  
    static boolean isAvailable = true;  
}
```

What is the result?

- A. Compilation fails.
- B. false true
- C. true false
- D. true true
- E. false false

**Answer:** C

**Explanation:**



```
Console 15 ✘ Console 16 ✘  
true false  
Completed with exit code: 0
```

#### NEW QUESTION 10

Which statement is true about the switch statement?

- A. It must contain the default section.
- B. The break statement, at the end of each case block, is mandatory.
- C. Its case label literals can be changed at runtime.
- D. Its expression must evaluate to a single value.

**Answer:** D

#### NEW QUESTION 10

Given:

```
class A {  
    public void test () {  
        System.out.println ("A");  
    }  
}  
class B extends A {  
    public void test () {  
        System.out.println ("B");  
    }  
}  
public class C extends A {  
    public void test () {  
        System.out.println ("C");  
    }  
  
    public static void main (String [] args) {  
        A b1 = new A ();  
        A b2 = new C ();  
  
        b1 = (A) b2;           //line n1  
        A b3 = (B) b2;         //line n2  
        b1.test ();  
        b3.test ();  
    }  
}
```

What is the result?

- A. AB
- B. AC
- C. CC
- D. A ClassCastException is thrown only at line n1.
- E. A ClassCastException is thrown only at line n2.

**Answer:** B

#### NEW QUESTION 11

Given the code fragment:

```
public static void main(String[] args) {  
    ArrayList<Integer> points = new ArrayList<>();  
    points.add(1);  
    points.add(2);  
    points.add(3);  
    points.add(4);  
    points.add(null);  
    points.remove(1);  
    points.remove(null);  
    System.out.println(points);  
}
```

What is the result?

- A. A NullPointerException is thrown at runtime
- B. [1, 2, 4]
- C. [1, 2, 4, null]
- D. [1, 3, 4, null]
- E. [1, 3, 4]
- F. Compilation fails.

**Answer:** B

#### NEW QUESTION 12

Given the code fragment:

```
int n [] [] = {{1, 3}, {2, 4}};  
for (int i = n.length-1; i >= 0; i--) {  
    for (int y : n[i]) {  
        System.out.print (y);  
    }  
}
```

What is the result?

- A. 1324
- B. 2313
- C. 3142
- D. 4231

**Answer:** D

#### NEW QUESTION 13

Which two statements are true about Java byte code? (Choose two.)

- A. It can be serialized across network.
- B. It can run on any platform that has a Java compiler.
- C. It can run on any platform.
- D. It has ".java" extension.
- E. It can run on any platform that has the Java Runtime Environment.

**Answer:** AE

#### NEW QUESTION 14

This grid shows the state of a 2D array:

|   |   |   |
|---|---|---|
| 0 | 0 |   |
|   | X | 0 |
| X |   | X |

The grid is created with this code:

```
char[][] grid = new char[3][3];
grid[1][1] = 'X';
grid[0][0] = 'O';
grid[2][0] = 'X';
grid[0][1] = 'O';
grid[2][2] = 'X';
grid[1][2] = 'O';
//line n1
```

Which line of code, when inserted in place of //line n1, adds an X into the grid so that the grid contains three consecutive Xs?

- A. grid[2][1] = 'X';
- B. grid[3][2] = 'X';
- C. grid[3][1] = 'X';
- D. grid[2][3] = 'X';

**Answer:** D

#### NEW QUESTION 15

Given:

```
public class Fieldinit {
    char c;
    boolean b;
    float f;
    void printAll() {
        System.out.println ("c = " + c);
        System.out.println ("b = " + b);
        System.out.println ("f = " + f);
    }
    public static void main (String [] args) {
        FieldInit f = new FieldInit ();
        f.printAll ();
    }
}
```

What is the result?

**A**

```
c=
b = false
f = 0.0
```

**B**

```
c= null
b = true
f = 0.0
```

**C**

```
c=0
b = false
f = 0.0f
```

**D**

```
c= null
b = false
f = 0.0F
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

**NEW QUESTION 19**

Given:

```
class Patient {  
    String name;  
    public Patient (String name) {  
        this.name = name;  
    }  
}
```

And the code fragment:

```
8. public class Test {  
9.     public static void main (String [] args) {  
10.         List ps = new ArrayList ();  
11.         Patient p2 = new Patient ("Mike");  
12.         ps.add(p2);  
13.  
14.         // insert code here  
15.  
16.         if (f >= 0) {  
17.             System.out.print ("Mike Found");  
18.         }  
19.     }  
20. }
```

Which code fragment, when inserted at line 14, enables the code to print Mike Found?

**A**

```
int f = ps.indexOf (p2);
```

**B**

```
int f = ps.indexOf (Patient ("Mike") );
```

**C**

```
int f = ps.indexOf (new Patient "Mike") );
```

**D**

```
Patient p = new Patient("Mike");  
int f = ps.indexOf(p)
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A**NEW QUESTION 23**

Given the code fragment:

```
public class Employee {  
    String name;  
    boolean contract;  
    double salary;  
    Employee() {  
        // line n1  
    }  
    public String toString(){  
        return name + ":" + contract + ":" + salary;  
    }  
    public static void main(String[] args) {  
        Employee e = new Employee();  
        // line n2  
        System.out.print(e);  
    }  
}
```

Which two modifications, when made independently, enable the code to print Joe:true: 100.0? (Choose two.)

- A) Replace line n2 with:

```
e.name = "Joe";  
e.contract = true;  
e.salary = 100;
```

- B) Replace line n2 with:

```
this.name = "Joe";  
this.contract = true;  
this.salary = 100;
```

- C) Replace line n1 with:

```
this.name = new String("Joe");  
this.contract = new Boolean(true);  
this.salary = new Double(100);
```

- D) Replace line n1 with:

```
name = "Joe";  
contract = TRUE;  
salary = 100.0f;
```

- E) Replace line n1 with:

```
this("Joe", true, 100);
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** AC

#### NEW QUESTION 24

Which is true about the switch statement?

- A. Its expression can evaluate to a collection of values.
- B. The break statement, at the end of each case block, is optional.
- C. Its case label literals can be changed at runtime.
- D. It must contain the default section.

**Answer:** B

#### NEW QUESTION 25

Given the code fragment:

```
public static void main(String[] args) {  
    LocalDate date = LocalDate.of(2012, 01, 32);  
    date.plusDays(10);  
    System.out.println(date);  
}
```

What is the result?

- A. 2012-02-10
- B. 2012-02-11
- C. Compilation fails
- D. A DateTimeException is thrown at runtime.

**Answer:** D

**NEW QUESTION 30**

Given:

```
class X {  
    int i;  
    static int j;  
    public static void main(String[] args) {  
        X x1 = new X();  
        X x2 = new X();  
        x1.i = 3;  
        x1.j = 4;  
        x2.i = 5;  
        x2.j = 6;  
        System.out.println(  
            x1.i + " " +  
            x1.j + " " +  
            x2.i + " " +  
            x2.j);  
    }  
}
```

What is the result?

- A. 3 4 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 5 6

**Answer:** D**Explanation:**

```
3 6 5 6  
Completed with exit code: 0
```

**NEW QUESTION 33**

Given the code fragment:

```
abstract class Planet {  
    protected void revolve() {  
        //line n1  
    }  
  
    abstract void rotate();  
    //line n2  
}  
  
class Earth extends Planet {  
    void revolve() {  
        //line n3  
    }  
  
    protected void rotate() {  
        //line n4  
    }  
}
```

Which two modifications, made independently, enable the code to compile? (Choose two.)

- A. Make the method at line n1 public.
- B. Make the method at line n2 public.
- C. Make the method at line n3 public.
- D. Make the method at line n3 protected.
- E. Make the method at line n4 public.

**Answer:** CD**NEW QUESTION 34**

Given:

```
class Caller {  
    private void init () {  
        System.out.println("Initialized");  
    }  
  
    private void start () {  
        init();  
        System.out.println("Started");  
    }  
}  
  
public class TestCall {  
    public static void main(String[] args) {  
        Caller c = new Caller();  
        c.start(); // line n1  
        c.init(); // line n2  
    }  
}
```

What is the result?

- A. Compilation fails at line n1.
- B. InitializedStartedInitialized
- C. InitializedStarted
- D. Compilation fails at line n2.

**Answer:** D

#### NEW QUESTION 38

Given this class:

```
public class CheckingAccount {  
    public int amount;  
    //line n1  
}
```

And given this main method, located in another class:

```
public static void main(String[] args) {  
    CheckingAccount acct = new CheckingAccount();  
    //line n2  
}
```

Which three pieces of code, when inserted independently, set the value of amount to 100?

A

At line n1 insert:  
public CheckingAccount() {  
 amount = 100;  
}

B

At line n2 insert:  
this.amount = 100;

C

At line n2 insert:  
amount = 100;

D

At line n1 insert:  
public CheckingAccount() {  
 this.amount = 100;  
}

E

At line n2 insert:  
acct.amount = 100;

F

At line n1 insert:  
public CheckingAccount() {  
 acct.amount = 100;  
}

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

**Answer:** DE

#### NEW QUESTION 41

Given:

```
class Test {  
    int a1;  
  
    public static void doProduct(int a) {  
        a = a * a;  
    }  
  
    public static void doString(String s) {  
        s.concat(" " + s);  
    }  
  
    public static void main(String[] args) {  
        Test item = new Test();  
        item.a1 = 11;  
        String sb = "Hello";  
        Integer i = 10;  
        doProduct(i);  
        doString(sb);  
        doProduct(item.a1);  
        System.out.println(i + " " + sb + " " + item.a1);  
    }  
}
```

What is the result?

- A. 10 Hello Hello 11
- B. 10 Hello Hello 121
- C. 100 Hello 121
- D. 100 Hello Hello 121
- E. 10 Hello 11

**Answer:** E

#### NEW QUESTION 42

Given:

```
public class Triangle {  
    static double area;  
    int b = 2, h = 3;  
    public static void main(String[] args) {  
        double p, b, h; //line n1  
        if (area == 0) {  
            b = 3;  
            h = 4;  
            p = 0.5;  
            area = p * b * h; //line n2  
        }  
        System.out.println("Area is " + area);  
    }  
}
```

What is the result?

- A. Area is 6.0
- B. Area is 3.0
- C. Compilation fails at line n1
- D. Compilation fails at line n2.

**Answer:** D

#### NEW QUESTION 44

Which two code fragments cause a compilation error? (Choose two.)

- A. float flt = 100.00F;
- B. float flt = (float) 1\_11.00;
- C. Float flt = 100.00;
- D. double y1 = 203.22;float flt = y1;
- E. int y2 = 100;float flt = (float) y2 ;

**Answer:** AD

#### NEW QUESTION 49

Given:

```
class Test {  
    public static void main (String [] args) {  
        int numbers [ ];  
        numbers = new int [2];  
        numbers [0] = 10;  
        numbers [1] = 20;  
  
        numbers = new int [4];  
        numbers [2] = 30;  
        numbers [3] = 40;  
        for (int x : numbers) {  
            System.out.print (" " + x) ;  
        }  
    }  
}
```

What is the result?

- A. 10 20 30 40
- B. 0 0 30 40
- C. Compilation fails.
- D. An exception is thrown at runtime.

**Answer:** C

#### NEW QUESTION 51

What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?

- A. Encapsulation
- B. Inheritance
- C. Abstraction
- D. Instantiation
- E. Polymorphism

**Answer:** A

#### Explanation:

Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

#### NEW QUESTION 52

Given the code fragment:

```
int wd = 0;
String days[] = {"sun", "mon", "wed", "sat"};
for (String s:days) {
    switch (s) {
        case "sat":
        case "sun":
            wd -= 1;
            break;
        case "mon":
            wd++;
        case "wed":
            wd += 2;
    }
}
System.out.println(wd);
```

What is the result?

- A. 3
- B. 4
- C. -1
- D. Compilation fails.

**Answer:** A

#### NEW QUESTION 57

Given:

```
public class Test {
    int x, y;

    public Test(int x, int y) {
        initialize(x, y);
    }

    public void initialize(int x, int y) {
        this.x = x * x;
        this.y = y * y;
    }

    public static void main(String[] args) {
        int x = 3, y = 5;
        Test obj = new Test(x, y);
        System.out.println(x + " " + y);
    }
}
```

What is the result?

- A. Compilation fails.
- B. 3 5
- C. 0 0

D. 9 25

**Answer:** B**NEW QUESTION 62**

Given:

```
public class Test {
    public static void main(String[] args) {
        Test ts = new Test();
        System.out.print(isAvailable + " ");
        isAvailable= ts.doStuff();
        System.out.println(isAvailable);
    }
    public static boolean doStuff() {
        return !isAvailable;
    }
    static boolean isAvailable = false;
}
```

What is the result?

- A. Compilation fails.
- B. false true
- C. true false
- D. true true
- E. false false

**Answer:** B**NEW QUESTION 67**

Given:

```
class Student {
    String name;
    public Student(String name) {
        this.name = name;
    }
}

public class Test {
    public static void main(String[] args) {
        Student[] students = new Student[3];
        students[1] = new Student("Richard");
        students[2] = new Student("Donald");
        for (Student s : students) {
            System.out.println(" " + s.name);
        }
    }
}
```

What is the result?

- A. nullRichardDonald
- B. RichardDonald
- C. Compilation fails.
- D. An ArrayIndexOutOfBoundsException is thrown at runtime.
- E. A NullPointerException is thrown at runtime.

**Answer:** E**NEW QUESTION 69**

Which three are advantages of the Java exception mechanism? (Choose three.)

- A. Improves the program structure because the error handling code is separated from the normal program function
- B. Provides a set of standard exceptions that covers all possible errors
- C. Improves the program structure because the programmer can choose where to handle exceptions
- D. Improves the program structure because exceptions must be handled in the method in which they occurred
- E. Allows the creation of new exceptions that are customized to the particular program being created

**Answer:** ACE**NEW QUESTION 72**

Given this class:

```
public class Rectangle {  
    private double length;  
    private double height;  
    private double area;  
  
    public void setLength(double length) {  
        this.length = length;  
    }  
    public void setHeight(double height) {  
        this.height = height;  
    }  
    public void setArea() {  
        area = length*height;  
    }  
}
```

Which two changes would encapsulate this class and ensure that the area field is always equal to length \* height whenever the Rectangle class is used?

- A. Call the setArea method at the end of the setHeight method.
- B. Call the setArea method at the beginning of the setHeight method.
- C. Call the setArea method at the end of the setLength method.
- D. Call the setArea method at the beginning of the setLength method.
- E. Change the setArea method to private.
- F. Change the area field to public.

**Answer:** AE

#### NEW QUESTION 76

Which statement is true about the switch statement?

- A. It must contain the default section.
- B. The break statement, at the end of each case block, is optional.
- C. Its case label literals can be changed at runtime.
- D. Its expression must evaluate to a collection of values.

**Answer:** B

#### NEW QUESTION 81

Given:

```
class Caller {  
    private void init () {  
        System.out.println("Initialized");  
    }  
  
    private void start () {  
        init();  
        System.out.println("Started");  
    }  
}  
  
public class TestCall {  
    public static void main(String[] args) {  
        Caller c = new Caller();  
        c.start();  
        c.init();  
    }  
}
```

What is the result?

- A. An exception is thrown at runtime.
- B. InitializedStartedInitialized
- C. InitializedStarted
- D. Compilation fails.

**Answer:** D

#### NEW QUESTION 84

Given the code fragment:

```
3. public static void main(String[] args) {  
4.     int x = 6;  
5.     while (isAvailable(x)) {  
6.         System.out.print(x);  
7.     }  
8. }  
10.  
11. public static boolean isAvailable(int x) {  
12.     return --x > 0 ? true : false;  
13. }
```

Which modification enables the code to print 54321?

- A. Replace line 6 with System.out.print (--x);
- B. At line 7, insert x --;
- C. Replace line 5 with while (is Available(--x)) {
- D. Replace line 12 with return (x > 0) ? false : true;

**Answer:** C

#### **NEW QUESTION 87**

Which statement best describes encapsulation?

- A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects.
- B. Encapsulation ensures that classes can be designed so that their methods are inheritable.
- C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.
- D. Encapsulation ensures that classes can be designed so that if a method has an argument MyType x, any subclass of MyType can be passed to that method.

**Answer:** A

#### **NEW QUESTION 91**

Which two statements are true? (Choose two.)

- A. Error class is unextendable.
- B. Error class is extendable.
- C. Error is a RuntimeException.
- D. Error is an Exception.
- E. Error is a Throwable.

**Answer:** BC

#### **NEW QUESTION 95**

Which three statements are true about the structure of a Java class? (Choose three.)

- A. A public class must have a main method.
- B. A class can have only one private constructors.
- C. A method can have the same name as a field.
- D. A class can have overloaded static methods.
- E. The methods are mandatory components of a class.
- F. The fields need not be initialized before use.

**Answer:** ACE

#### **NEW QUESTION 97**

.....

## THANKS FOR TRYING THE DEMO OF OUR PRODUCT

Visit Our Site to Purchase the Full Set of Actual 1z0-808 Exam Questions With Answers.

We Also Provide Practice Exam Software That Simulates Real Exam Environment And Has Many Self-Assessment Features. Order the 1z0-808 Product From:

<https://www.2passey.com/dumps/1z0-808/>

## Money Back Guarantee

### 1z0-808 Practice Exam Features:

- \* 1z0-808 Questions and Answers Updated Frequently
- \* 1z0-808 Practice Questions Verified by Expert Senior Certified Staff
- \* 1z0-808 Most Realistic Questions that Guarantee you a Pass on Your First Try
- \* 1z0-808 Practice Test Questions in Multiple Choice Formats and Updates for 1 Year

➤ **Vendor: Oracle**

➤ **Exam Code: 1Z0-808**

➤ **Exam Name: Java SE 8 Programmer I**

➤ **Question 1 -- Question 20**

[\*\*Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps\*\*](#)

**QUESTION 1**

Given:

```
interface Readable {  
    public void readBook();  
    public void setBookMark();  
}  
  
abstract class Book implements Readable { // line n1  
    public void readBook() {}  
    // line n2  
}  
  
class EBook extends Book { // line n3  
    public void readBook() {}  
    // line n4  
}
```

Which option enables the code to compile?

- A) Replace the code fragment at line n1 with:  

```
class Book implements Readable {
```
  - B) At line n2 insert:  

```
public abstract void setBookMark();
```
  - C) Replace the code fragment at line n3 with:  

```
abstract class EBook extends Book {
```
  - D) At line n4 insert:  

```
public void setBookMark() { }
```
- A. Option A  
B. Option B  
C. Option C  
D. Option D

**Answer:** D

## QUESTION 2

Given the code fragment:

```
public static void main(String[] args) {
    List<String> names = new ArrayList<>();
    names.add("Robb");
    names.add("Bran");
    names.add("Rick");
    names.add("Bran");

    if (names.remove("Bran")) {
        names.remove("Jon");
    }
    System.out.println(names);
}
```

What is the result?

- A. [Robb, Rick, Bran]
- B. [Robb, Rick]
- C. [Robb, Bran, Rick, Bran]
- D. An exception is thrown at runtime.

**Answer:** A

**Explanation:**

After adding elements to names we have a list with four elements and element "Bran" repeated.

After removing element "Bran" we have a list with three elements [Robb, Rick, Bran].

remove method removes the first occurrence of the specified element from this list, if it is present.

If the list does not contain the element, it is unchanged.

<https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html#remove-java.lang.Object->

**QUESTION 3**

Given:

```
class A {  
    public A(){  
        System.out.print("A ");  
    }  
}  
  
class B extends A{  
    public B(){  
        System.out.print("B ");  
    }  
}  
  
class C extends B{  
  
    public C(){  
        System.out.print("C ");  
    }  
    public static void main(String[] args) {  
        C c = new C();  
    }  
}
```

What is the result?

- A. C B A
- B. C
- C. A B C
- D. Compilation fails at line n1 and line n2

**Answer: C**

**QUESTION 4**

Given:

```
class X {
    static int i;
    int j;
    public static void main(String[] args) {
        X x1 = new X();
        X x2 = new X();
        x1.i = 3;
        x1.j = 4;
        x2.i = 5;
        x2.j = 6;
        System.out.println(
            x1.i + " " +
            x1.j + " " +
            x2.i + " " +
            x2.j);
    }
}
```

What is the result?

- A. 3 4 5 6
- B. 3 4 3 6
- C. 5 4 5 6
- D. 3 6 4 6

**Answer: C**

**Explanation:**

Since variable i is static, it is shared by all instances of X. When code executes x2.i = 5, x1.i = 5 too.

Since variable j isn't static, each instance of X has its own copy of j.

#### QUESTION 5

Given the code fragment:

```
1. public class Test {
2.     public static void main(String[] args) {
3.         /* insert code here */
4.         array[0]=10;
5.         array[1]=20;
6.         System.out.print(array[0]+":"+array[1]);
7.     }
8. }
```

Which code fragment, when inserted at line 3, enables the code to print 10:20?

- A. int[] array = new int[2];
- B. int[] array;  
array = int[2];
- C. int array = new int[2];
- D. int array [2] ;

**Answer:** B

**QUESTION 6**

Given the code fragment:

```
public static void main(String[] args) {  
    String[] arr = {"A", "B", "C", "D"};  
    for (int i = 0; i < arr.length; i++) {  
        System.out.print(arr[i] + " ");  
        if (arr[i].equals("C")) {  
            continue;  
        }  
        System.out.println("Work done");  
        break;  
    }  
}
```

What is the result?

- A. A B C Work done
- B. A B C D Work done
- C. A Work done
- D. Compilation fails

**Answer:** C

**QUESTION 7**

Which three are advantages of the Java exception mechanism?

- A. Improves the program structure because the error handling code is separated from the normal program function
- B. Provides a set of standard exceptions that covers all the possible errors
- C. Improves the program structure because the programmer can choose where to handle exceptions
- D. Improves the program structure because exceptions must be handled in the method in which they occurred
- E. Allows the creation of new exceptions that are tailored to the particular program being created

**Answer:** ACE

**Explanation:**

B is false. Standard exceptions not cover all possible errors.

D is false. Exceptions don't have to be handled in the method in which they occurred.

**QUESTION 8**

Given the code from the Greeting.Java file:

```
public class Greeting {  
    public static void main(String[] args) {  
        System.out.println("Hello " + args[0]);  
    }  
}
```

Which set of commands prints Hello Duke in the console?

- A) javac Greeting  
java Greeting Duke
  - B) javac Greeting.java Duke  
java Greeting
  - C) javac Greeting.java  
java Greeting Duke
  - D) javac Greeting.java  
java Greeting.class Duke
- A. Option A  
B. Option B  
C. Option C  
D. Option D

**Answer:** C

**Explanation:**

Source code file names must have .java suffixes to compile with javac

We interpret or run the program with “java <class name without suffix> arguments”

<http://docs.oracle.com/javase/8/docs/technotes/tools/windows/javac.html>

<http://docs.oracle.com/javase/8/docs/technotes/tools/windows/java.html>

### QUESTION 9

Given:

```
class Alpha {  
    int ns;  
    static int s;  
    Alpha(int ns) {  
        if (s < ns) {  
            s = ns;  
            this.ns = ns;  
        }  
    }  
    void doPrint() {  
        System.out.println("ns = " + ns + " s = " + s);  
    }  
}
```

And,

```
public class TestA {  
    public static void main(String[] args) {  
        Alpha ref1 = new Alpha(50);  
        Alpha ref2 = new Alpha(125);  
        Alpha ref3 = new Alpha(100);  
        ref1.doPrint();  
        ref2.doPrint();  
        ref3.doPrint();  
    }  
}
```

What is the result?

- A) ns = 50 s = 125  
ns = 125 s = 125  
ns = 100 s = 125
- B) ns = 50 s = 125  
ns = 125 s = 125  
ns = 0 s = 125
- C) ns = 50 s = 50  
ns = 125 s = 125  
ns = 100 s = 100
- D) ns = 50 s = 50  
ns = 125 s = 125  
ns = 0 s = 125

A. Option A

B. Option B

- C. Option C
- D. Option D

**Answer:** B

**Explanation:**

After ref1 is instantiated, ref1.ns = 50 and s = 50

After ref2 is instantiated, ref2.ns = 125 and s = 125

After ref3 is instantiated, ref3.ns = 0 and s = 125

#### QUESTION 10

Given the code fragment:

```
public static void main(String[] args) {  
    int ii = 0;  
    int jj = 7;  
    for (ii = 0; ii < jj - 1; ii = ii + 2) {  
        System.out.print(ii + " ");  
    }  
}
```

What is the result?

- A. 2 4
- B. 0 2 4 6
- C. 0 2 4
- D. Compilation fails

**Answer:** C

#### QUESTION 11

Given the code fragment:

```
LocalDate date1 = LocalDate.now();  
LocalDate date2 = LocalDate.of(2014, 6, 20);  
LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);  
System.out.println("date1 = " + date1);  
System.out.println("date2 = " + date2);  
System.out.println("date3 = " + date3);
```

Assume that the system date is June 20, 2014. What is the result?

- A) date1 = 2014-06-20  
date2 = 2014-06-20  
date3 = 2014-06-20
- B) date1 = 06/20/2014  
date2 = 2014-06-20  
date3 = Jun 20, 2014
- C) Compilation fails.
- D) A DateParseException is thrown at runtime.

- A. Option A

- B. Option B
- C. Option C
- D. Option D

**Answer:** A

**Explanation:**

I've run the following code without any problem

```
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
public class Main {
    public static void main(String[] args) {
        LocalDate date1 = LocalDate.now();
        LocalDate date2 = LocalDate.of(2014, 6, 20);
        LocalDate date3 = LocalDate.parse("2014-06-20", DateTimeFormatter.ISO_DATE);
        System.out.println("date 1 = " + date1);
        System.out.println("date 2 = " + date2);
        System.out.println("date 3 = " + date3);
    }
}
```

The output is

date 1 = 2015-09-05 (because run today, but problem statement says we must assume that the system data is June 20, 2014)  
date 2 = 2014-06-20  
date 3 = 2014-06-20

### QUESTION 12

Given the code fragment:

```
7. StringBuilder sb1 = new StringBuilder("Duke");
8. String str1 = sb1.toString();
9. // insert code here
10. System.out.print(str1 == str2);
```

Which code fragment, when inserted at line 9, enables the code to print true?

- A. String str2 = str1;
- B. String str2 = new String (str1);
- C. String str2 = sb1. toString ();
- D. String str2 = "Duke";

**Answer:** A

**Explanation:**

Operator == checks if two things are EXACTLY the same thing, not if they have the same content

### QUESTION 13

Given the code fragment:

```
public class Test {  
  
    static int count = 0;  
    int i = 0;  
  
    public void changeCount() {  
        while (i < 5) {  
            i++;  
            count++;  
        }  
    }  
  
    public static void main(String[] args) {  
        Test check1 = new Test();  
        Test check2 = new Test();  
        check1.changeCount();  
        check2.changeCount();  
        System.out.print(check1.count + " : " + check2.count);  
    }  
}
```

What is the result?

- A. 10 : 10
- B. 5 : 5
- C. 5 : 10
- D. Compilation fails

**Answer: A**

**Explanation:**

The variable i is local to all instances of class Test so each time we create an instance, i=0 and the loop add 5 to count.

The variable count (static) is global to all instances of class Test and all instances share the same variable. It's been initialized only once to zero and retains its value between the calls to changeCount. Since we call two times the method changeCount, the final result is 10 : 10

#### **QUESTION 14**

Given the code fragment:

```
public static void main(String[] args) {  
    double discount = 0;  
    int qty = Integer.parseInt(args[0]);  
    //line n1;  
}
```

And given the requirements:

- If the value of the qty variable is greater than or equal to 90, discount = 0.5
  - If the value of the qty variable is between 80 and 90, discount = 0.2
- Which two code fragments can be independently placed at line n1 to meet the requirements?

- A) if (qty >= 90) { discount = 0.5; }  
    if (qty > 80 && qty < 90) { discount = 0.2; }
  - B) discount = (qty >= 90) ? 0.5 : 0;  
    discount = (qty > 80) ? 0.2 : 0;
  - C) discount = (qty >= 90) ? 0.5 : (qty > 80)? 0.2 : 0;
  - D) if (qty > 80 && qty < 90) {  
        discount = 0.2;  
    } else {  
        discount = 0;  
    }  
    if (qty >= 90) {  
        discount = 0.5;  
    } else {  
        discount = 0;  
    }
  - E) discount = (qty > 80) ? 0.2 : (qty >= 90) ? 0.5 : 0;
- A. Option A  
B. Option B  
C. Option C  
D. Option D  
E. Option E

**Answer:** AC

**QUESTION 15**

Given:

```
public class Test {  
  
    public static void main(String[] args) {  
        if (args[0].equals("Hello") ? false : true) {  
            System.out.println("Success");  
        } else {  
            System.out.println("Failure");  
        }  
    }  
}
```

And given the commands:

javac Test.java

Java Test Hello

What is the result?

- A. Success

- B. Failure
- C. Compilation fails.
- D. An exception is thrown at runtime

**Answer:** B

#### **QUESTION 16**

Which three statements describe the object-oriented features of the Java language?

- A. Objects cannot be reused.
- B. A subclass can inherit from a superclass.
- C. Objects can share behaviors with other objects.
- D. A package must contain more than one class.
- E. Object is the root class of all other objects.
- F. A main method must be declared in every class.

**Answer:** BCE

**Explanation:**

<https://docs.oracle.com/javase/tutorial/java/landl/subclasses.html>

<http://www.artima.com/objectsandjava/webuscript/PolymorphismInterfaces1.html>

#### **QUESTION 17**

Given the following code:

```
public static void main(String[] args){  
    String[] planets = {"Mercury", "Venus", "Earth", "Mars"};  
  
    System.out.println(planets.length);  
    System.out.println(planets[1].length());  
}
```

What is the output?

- A. 4  
4
- B. 3  
5
- C. 4  
7
- D. 5  
4
- E. 4  
5
- F. 4  
21

**Answer:** E

#### **QUESTION 18**

You are developing a banking module.

You have developed a class named ccMask that has a maskcc method.

Given the code fragment:

```
class CCmask {
    public static String maskCC(String creditCard) {
        String x = "XXXX-XXXX-XXXX-";
        //line n1
    }

    public static void main(String[] args) {
        System.out.println(maskCC("1234-5678-9101-1121"));
    }
}
```

You must ensure that the maskcc method returns a string that hides all digits of the credit card number except the four last digits (and the hyphens that separate each group of four digits). Which two code fragments should you use at line n1, independently, to achieve this requirement?

- A) `StringBuilder sb = new StringBuilder(creditCard);
 sb.substring(15, 19);
 return x + sb;`
- B) `return x + creditCard.substring(15, 19);`
- C) `StringBuilder sb = new StringBuilder(x);
 sb.append(creditCard, 15, 19);
 return sb.toString();`
- D) `StringBuilder sb = new StringBuilder(creditCard);
 StringBuilder s = sb.insert(0, x);
 return s.toString();`

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** BC

#### **QUESTION 19**

Given the code fragment:

```
public class App {  
    public static void main(String[] args) {  
        String str1 = "Java";  
        String str2 = new String("java");  
        //line n1  
        {  
            System.out.println("Equal");  
        } else {  
            System.out.println("Not Equal");  
        }  
    }  
}
```

Which code fragment, when inserted at line n1, enables the App class to print Equal?

- A) String str3 = str2;  
if (str1 == str3)
  - B) if (str1.equalsIgnoreCase(str2))
  - C) String str3 = str2;  
if (str1.equals(str3))
  - D) if (str1.toLowerCase() == str2.toLowerCase())
- A. Option A  
B. Option B  
C. Option C  
D. Option D

**Answer:** A

#### **QUESTION 20**

Given:

```
public class SumTest {  
  
    public static void doSum(Integer x, Integer y) {  
        System.out.println("Integer sum is " + (x + y));  
    }  
  
    public static void doSum(double x, double y) {  
        System.out.println("double sum is " + (x + y));  
    }  
  
    public static void doSum(float x, float y) {  
        System.out.println("float sum is " + (x + y));  
    }  
  
    public static void doSum(int x, int y) {  
        System.out.println("int sum is " + (x + y));  
    }  
  
    public static void main(String[] args) {  
        doSum(10, 20);  
        doSum(10.0, 20.0);  
    }  
}
```

What is the result?

- A) int sum is 30  
float sum is 30.0
- B) int sum is 30  
double sum is 30
- C) Integer sum is 30  
double sum is 30.0
- D) Integer sum is 30  
float sum is 30.0

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** B

**Explanation:**

int is a primitive type and Integer is an object with an int. When we call doSum(10, 20), we are calling doSum(int, int).



Leader of IT Certifications

[Free VCE and PDF Exam Dumps from PassLeader](#)

By default, Java uses double to represent its floating point literals. When we call doSum(10.0, 20.0), we are calling doSum(double, double).

[Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps](#)

➤ **Vendor: Oracle**

➤ **Exam Code: 1Z0-808**

➤ **Exam Name: Java SE 8 Programmer I**

➤ **Question 21 -- Question 40**

[\*\*Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps\*\*](#)

**QUESTION 21**

Given the code fragment:

```
String[] strs = new String[2];
int idx = 0;
for (String s : strs) {
    strs[idx].concat(" element " + idx);
    idx++;
}
for (idx = 0; idx < strs.length; idx++) {
    System.out.println(strs[idx]);
}
```

What is the result?

- A. Element 0  
Element 1
- B. Null element 0  
Null element 1
- C. Null  
Null
- D. A NullPointerException is thrown at runtime.

**Answer:** D

**QUESTION 22**

Given:

```
class Vehicle {  
    int x;  
    Vehicle(){  
        this(10); // line n1  
    }  
    Vehicle(int x) {  
        this.x = x;  
    }  
}  
  
class Car extends Vehicle {  
    int y;  
    Car() {  
        super();  
        this(20); // line n2  
    }  
    Car(int y) {  
        this.y = y;  
    }  
    public String toString() {  
        return super.x + ":" + this.y;  
    }  
}
```

And given the code fragment:

And given the code fragment:

```
Vehicle y = new Car();  
System.out.println(y);
```

What is the result?

- A. 10:20
- B. 0:20
- C. Compilation fails at line n1
- D. Compilation fails at line n2

**Answer: D**

**Explanation:**

`this()` and `super()` can't be used in the same constructor

Here is a good reference for the question

<http://stackoverflow.com/questions/10381244/why-can-t-this-and-super-both-be-used-together-in-a-constructor>

### QUESTION 23

Given the definitions of the `MyString` class and the `Test` class:

MyString.java:

```
package p1;
class MyString {
    String msg;
    MyString(String msg) {
        this.msg = msg;
    }
}
```

Test.java:

```
package p1;
public class Test {
    public static void main(String[] args) {
        System.out.println("Hello " + new StringBuilder("Java SE 8"));
        System.out.println("Hello " + new MyString("Java SE 8"));
    }
}
```

What is the result?

- A) Hello Java SE 8  
Hello Java SE 8
- B) Hello java.lang.StringBuilder@<<hashcode1>>  
Hello p1.MyString@<<hashcode2>>
- C) Hello Java SE 8  
Hello p1.MyString@<<hashcode>>
- D) Compilation fails at the Test class.

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer: C**

**QUESTION 24**

Given the code fragment:

```
3. public static void main(String[] args) {
4.     int iVar = 100;
5.     float fVar = 100.100f;
6.     double dVar = 123;
7.     iVar = fVar;
8.     fVar = iVar;
9.     dVar = fVar;
10.    fVar = dVar;
11.    dVar = iVar;
12.    iVar = dVar;
13. }
```

Which three lines fail to compile?

- A. Line 7
- B. Line 8
- C. Line 9
- D. Line 10
- E. Line 11
- F. Line 12

**Answer:** ADF

**Explanation:**

See "Assignment Compatibility" at  
[http://docstore.mik.ua/orelly/java/langref/ch04\\_13.htm](http://docstore.mik.ua/orelly/java/langref/ch04_13.htm)

### QUESTION 25

Given:

MainTest.java:

```
public class MainTest {

    public static void main(int[] args) {
        System.out.println("int main " + args[0]);
    }
    public static void main(Object[] args) {
        System.out.println("Object main " + args[0]);
    }
    public static void main(String[] args) {
        System.out.println("String main " + args[0]);
    }
}
```

and commands:

```
javac MainTest.java
java MainTest 1 2 3
```

What is the result?

- A. int main 1
- B. Object main 1
- C. String main 1
- D. Compilation fails
- E. An exception is thrown at runtime

**Answer:** C

**Explanation:**

All methods have the same name but different signature since the parameters are different. There is no problem with that.

JVM will call the method with signature “public static void main(String[] args)”

<https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html>

### QUESTION 26

Given the code fragment:

```
int num[][] = new int[1][3];
for (int i = 0; i < num.length; i++) {
    for (int j = 0; j < num[i].length; j++) {
        num[i][j] = 10;
    }
}
```

Which option represents the state of the num array after successful completion of the outer loop?

- A) num[0][0]=10  
num[0][1]=10  
num[0][2]=10
- B) num[0][0]=10  
num[1][0]=10  
num[2][0]=10
- C) num[0][0]=10  
num[0][1]=0  
num[0][2]=0
- D) num[0][0]=10  
num[0][1]=10  
num[0][2]=10  
num[0][3]=10  
num[1][0]=0  
num[1][1]=0  
num[1][2]=0  
num[1][3]=0

- A. Option A
- B. Option B
- C. Option C

D. Option D

**Answer:** A

**Explanation:**

At first look we can exclude option D because the number of elements in the array is 3, the result of multiplying the two array dimensions 1 x 3.

We can run the code

```
public class Main {  
    public static void main(String[] args) {  
        int num[][] = new int[1][3];  
        for (int i=0; i<num.length; i++) {  
            for (int j=0; j<num[i].length; j++) {  
                num[i][j] = 10;  
            System.out.println("num[" + i + "][" + j + "] = " + num[i][j]);  
        }  
    }  
}
```

the output is

```
num[0][0]= 10  
num[0][1]= 10  
num[0][2]= 10
```

### QUESTION 27

Given the code fragment:

```
public class Person {  
    String name;  
    int age = 25;  
  
    public Person(String name) {  
        this(); //line n1  
        setName(name);  
    }  
  
    public Person(String name, int age) {  
        Person(name); //line n2  
        setAge(age);  
    }  
  
    //setter and getter methods go here  
  
    public String show() {  
        return name + " " + age + " " + number ;  
    }  
    public static void main(String[] args) {  
        Person p1 = new Person("Jesse");  
        Person p2 = new Person("Walter", 52);  
        System.out.println(p1.show());  
        System.out.println(p2.show());  
    }  
}
```

What is the result?

- A. Jesse 25  
Walter 52
- B. Compilation fails only at line n1
- C. Compilation fails only at line n2
- D. Compilation fails at both line n1 and line n2

**Answer:** D

**Explanation:**

At line n1, Person class hasn't any constructor without arguments.

At line n2, there isn't any method Person. If we want to call the constructor that should be "this(name)".

### QUESTION 28

Given the following code for a Planet object:

```
public class Planet {  
    public String name;  
    public int moons;  
  
    public Planet(String name, int moons) {  
        this.name = name;  
        this.moons = moons;  
    }  
}
```

And the following main method:

```
public static void main(String[] args) {  
    Planet[] planets = {  
        new Planet("Mercury", 0),  
        new Planet("Venus", 0),  
        new Planet("Earth", 1),  
        new Planet("Mars", 2)  
    };  
  
    System.out.println(planets);  
    System.out.println(planets[2]);  
    System.out.println(planets[2].moons);  
}
```

What is the output?

- A) planets  
Earth  
1
  - B) [LPlanets.Planet;@15db9742  
Earth  
1
  - C) [LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
1
  - D) [LPlanets.Planet;@15db9742  
Planets.Planet@6d06d69c  
[LPlanets.Moon;@7852e922
  - E) [LPlanets.Planet;@15db9742  
Venus  
0
- A. Option A  
B. Option B  
C. Option C  
D. Option D  
E. Option E

**Answer:** C

#### **QUESTION 29**

You are asked to develop a program for a shopping application, and you are given the following information:

- The application must contain the classes Toy, EduToy, and consToy.
- The Toy class is the superclass of the other two classes.
- The int calculatePrice (Toy t) method calculates the price of a toy.
- The void printToy (Toy t) method prints the details of a toy.

Which definition of the Toy class adds a valid layer of abstraction to the class hierarchy?

- A) public abstract class Toy{  
    public abstract int calculatePrice(Toy t);  
    public void printToy(Toy t) { /\* code goes here \*/ }  
}
  - B) public abstract class Toy {  
    public int calculatePrice(Toy t) ;  
    public void printToy(Toy t) ;  
}
  - C) public abstract class Toy {  
    public int calculatePrice(Toy t);  
    public final void printToy(Toy t){ /\* code goes here \*/ }  
}
  - D) public abstract class Toy {  
    public abstract int calculatePrice(Toy t) { /\* code goes here \*/ }  
    public abstract void printToy(Toy t) { /\* code goes here \*/ }  
}
- A. Option A  
B. Option B  
C. Option C  
D. Option D

**Answer:** B**Explanation:**<https://docs.oracle.com/javase/tutorial/java/landl/abstract.html>**QUESTION 30**

Given the following code:

```
int[] intArr = {15, 30, 45, 60, 75};  
intArr[2] = intArr[4];  
intArr[4] = 90;
```

What are the values of each element in intArr after this code has executed?

- A. 15, 60, 45, 90, 75  
B. 15, 90, 45, 90, 75  
C. 15, 30, 75, 60, 90  
D. 15, 30, 90, 60, 90  
E. 15, 4, 45, 60, 90

**Answer:** C**QUESTION 31**

Given the following array:

```
int[] intArr = {8, 16, 32, 64, 128};
```

Which two code fragments, independently, print each element in this array?

- A) 

```
for (int i : intArr) {
    System.out.print(intArr[i] + " ");
}
```
- B) 

```
for (int i : intArr) {
    System.out.print(i + " ");
}
```
- C) 

```
for (int i=0 : intArr) {
    System.out.print(intArr[i] + " ");
    i++;
}
```
- D) 

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(i + " ");
}
```
- E) 

```
for (int i=0; i < intArr.length; i++) {
    System.out.print(intArr[i] + " ");
}
```
- F) 

```
for (int i; i < intArr.length; i++) {
    System.out.print(intArr[i] + " ");
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E
- F. Option F

**Answer:** BE

#### **QUESTION 32**

Given the content of three files:

A.java:

```
public class A {  
    public void a() {}  
    int a;  
}
```

B.java:

```
public class B {  
    private int doStuff() {  
        private int x = 100;  
        return x++;  
    }  
}
```

C.java:

```
import java.io.*;  
package p1;  
class A {  
    public void main(String fileName) throws IOException {}  
}
```

Which statement is true?

- A. Only the A.java file compiles successfully.
- B. Only the B.java file compiles successfully.
- C. Only the C.java file compiles successfully.
- D. The A.java and B.java files compile successfully.
- E. The B.java and C.java files compile successfully.
- F. The A.java and C.java files compile successfully.

**Answer: A**

**Explanation:**

Class B doesn't compile because we can't use access modifiers (private) inside methods.

Class C doesn't compile because if the class is part of a package (p1), the package statement must be the first line in the source code file, before any import statements (java.io.\*) that may be present.

### QUESTION 33

Given the code fragment:

```
int[] array = {1, 2, 3, 4, 5};
```

And given the requirements:

1. Process all the elements of the array in the order of entry.
2. Process all the elements of the array in the reverse order of entry.
3. Process alternating elements of the array in the order of entry.

Which two statements are true?

- A. Requirements 1, 2, and 3 can be implemented by using the enhanced for loop.
- B. Requirements 1, 2, and 3 can be implemented by using the standard for loop.
- C. Requirements 2 and 3 CANNOT be implemented by using the standard for loop.

- D. Requirement 1 can be implemented by using the enhanced for loop.
- E. Requirement 3 CANNOT be implemented by using either the enhanced for loop or the standard for loop.

**Answer:** DE

#### **QUESTION 34**

Given:

```
public class TestScope {  
    public static void main(String[] args) {  
        int var1 = 200;  
        System.out.print(doCalc(var1));  
        System.out.print(" "+var1);  
    }  
    static int doCalc(int var1){  
        var1 = var1 * 2;  
        return var1;  
    }  
}
```

What is the result?

- A. 400 200
- B. 200 200
- C. 400 400
- D. Compilation fails.

**Answer:** A

#### **QUESTION 35**

Given the following class declarations:

- public abstract class Animal
- public interface Hunter
- public class Cat extends Animal implements Hunter
- public class Tiger extends Cat

Which answer fails to compile?

- A) `ArrayList<Animal> myList = new ArrayList<>();  
myList.add(new Tiger());`
  - B) `ArrayList<Hunter> myList = new ArrayList<>();  
myList.add(new Cat());`
  - C) `ArrayList<Hunter> myList = new ArrayList<>();  
myList.add(new Tiger());`
  - D) `ArrayList<Tiger> myList = new ArrayList<>();  
myList.add(new Cat());`
  - E) `ArrayList<Animal> myList = new ArrayList<>();  
myList.add(new Cat());`
- A. Option A  
B. Option B  
C. Option C  
D. Option D  
E. Option E

**Answer:** D

**Explanation:**

Cat cannot be converted to Tiger.

One Tiger is a Cat but one Cat isn't a Tiger.

### QUESTION 36

Which statement is true about Java byte code?

- A. It can run on any platform.
- B. It can run on any platform only if it was compiled for that platform.
- C. It can run on any platform that has the Java Runtime Environment.
- D. It can run on any platform that has a Java compiler.
- E. It can run on any platform only if that platform has both the Java Runtime Environment and a Java compiler.

**Answer:** C

**Explanation:**

We are talking about byte code so the Java program has been compiled.

The question ask for what we need to run the byte code.

[https://www.java.com/en/download/faq/whatis\\_java.xml](https://www.java.com/en/download/faq/whatis_java.xml)

[http://www.researchgate.net/post/Run\\_Java\\_Application\\_Without\\_Installing\\_Java\\_Runtime](http://www.researchgate.net/post/Run_Java_Application_Without_Installing_Java_Runtime)

### QUESTION 37

Given:

```
public class MarkList {  
    int num;  
    public static void graceMarks(MarkList obj4) {  
        obj4.num += 10;  
    }  
    public static void main(String[] args) {  
        MarkList obj1 = new MarkList();  
        MarkList obj2 = obj1;  
        MarkList obj3 = null;  
        obj2.num = 60;  
        graceMarks(obj2);  
    }  
}
```

How many MarkList instances are created in memory at runtime?

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

**Answer:** A

**Explanation:**

Only the statement “MarList obj1 = new MarList();” creates an instance of MarList.

### QUESTION 38

Given:

```
public class Triangle {  
    static double area;  
    int b = 2, h = 3;  
    public static void main(String[] args) {  
        double p, b, h; //line n1  
        if (area == 0) {  
            b = 3;  
            h = 4;  
            p = 0.5;  
        }  
        area = p * b * h; //line n2  
        System.out.println("Area is " + area);  
    }  
}
```

What is the result?

- A. Area is 6.0
- B. Area is 3.0
- C. Compilation fails at line n1
- D. Compilation fails at line n2.

Answer: D

**QUESTION 39**

Given the code fragment:

```
public class Test {
    public static void main(String[] args) {
        //line n1
        switch (x) {
            case 1:
                System.out.println("One");
                break;
            case 2:
                System.out.println("Two");
                break;
        }
    }
}
```

Which three code fragments can be independently inserted at line n1 to enable the code to print one?

- A. Byte x = 1;
- B. short x = 1;
- C. String x = "1";
- D. Long x = 1;
- E. Double x = 1;
- F. Integer x = new Integer ("1");

Answer: ABF

**QUESTION 40**

Given:

```
public class App {

    public static void main(String[] args) {
        Boolean[] bool = new Boolean[2];

        bool[0] = new Boolean(Boolean.parseBoolean("true"));
        bool[1] = new Boolean(null);

        System.out.println(bool[0] + " " + bool[1]);
    }
}
```

What is the result?

- A. True false
- B. True null
- C. Compilation fails
- D. A NullPointerException is thrown at runtime

**Answer:** A

**Explanation:**

With the statement “bool[1] = new Boolean(null);” we are creating a wrapped Boolean object with value null.

Java evaluates it to false since it cannot evaluate to true.

[Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps](#)

➤ **Vendor: Oracle**

➤ **Exam Code: 1Z0-808**

➤ **Exam Name: Java SE 8 Programmer I**

➤ **Question 41 -- Question 60**

[Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps](#)

**QUESTION 41**

Given the following code for the classes MyException and Test:

```
public class MyException extends RuntimeException {}  
  
public class Test {  
    public static void main(String[] args) {  
        try {  
            method1();  
        }  
        catch (MyException ne) {  
            System.out.print("A");  
        }  
    }  
    public static void method1() { // line n1  
        try {  
            throw Math.random() > 0.5 ?new MyException() :new RuntimeException();  
        }  
        catch (RuntimeException re) {  
            System.out.print("B");  
        }  
    }  
}
```

What is the result?

- A. A
- B. B
- C. Either A or B
- D. A B
- E. A compile time error occurs at line n1

**Answer:** B

**Explanation:**

"catch (RuntimeException re)" always catches a RuntimeException.

**QUESTION 42**

Given:

```
public class App {  
  
    String myStr = "7007";  
  
    public void doStuff(String str) {  
        int myNum = 0;  
        try {  
            String myStr = str;  
            myNum = Integer.parseInt(myStr);  
        } catch (NumberFormatException ne) {  
            System.err.println("Error");  
        }  
        System.out.println(  
            "myStr: " + myStr + ", myNum: " + myNum);  
    }  
  
    public static void main(String[] args) {  
        App obj = new App();  
        obj.doStuff("9009");  
    }  
}
```

What is the result?

- A. myStr: 9009, myNum: 9009
- B. myStr: 7007, myNum: 7007
- C. myStr: 7007, myNum: 9009
- D. Compilation fails

**Answer:** C

**QUESTION 43**

Which two are benefits of polymorphism?

- A. Faster code at runtime
- B. More efficient code at runtime
- C. More dynamic code at runtime
- D. More flexible and reusable code
- E. Code that is protected from extension by other classes

**Answer:** BD

**Explanation:**

<https://www.cs.princeton.edu/courses/archive/fall98/cs441/mainus/node5.html>

**QUESTION 44**

Given the code fragment:

```
int nums1[] = new int[3];
int nums2[] = {1, 2, 3, 4, 5};
nums1 = nums2;
for (int x : nums1){
    System.out.print(x + ":");
}
```

What is the result?

- A. 1:2:3:4:5:
- B. 1:2:3:
- C. Compilation fails.
- D. An ArrayOutOfBoundsException is thrown at runtime.

**Answer: A**

**QUESTION 45**

Given:

```
public class Product {
    int id;
    String name;
    public Product(int id, String name) {
        this.id = id;
        this.name = name;
    }
}
```

And given the code fragment:

```
4. Product p1 = new Product(101, "Pen");
5. Product p2 = new Product(101, "Pen");
6. Product p3 = p1;
7. boolean ans1 = p1 == p2;
8. boolean ans2 = p1.name.equals(p2.name);
9. System.out.print(ans1 + ":" + ans2);
```

What is the result?

- A. true:true
- B. true:false
- C. false:true
- D. false:false

**Answer: C**

**QUESTION 46**

Given the following classes:

```
public class Employee {  
    public int salary;  
}  
  
public class Manager extends Employee {  
    public int budget;  
}  
  
public class Director extends Manager {  
    public int stockOptions;  
}
```

And given the following main method:

```
public static void main(String[] args) {  
    Employee employee = new Employee();  
    Manager manager = new Manager();  
    Director director = new Director();  
    //line n1  
}
```

Which two options fail to compile when placed at line n1 of the main method?

- A. employee.salary = 50\_000;
- B. director.salary = 80\_000;
- C. employee.budget = 200\_000;
- D. manager.budget = 1\_000\_000;
- E. manager.stockOption = 500;
- F. director.stockOptions = 1\_000;

**Answer: CE**

**Explanation:**

- C. budget is not a member of class employee.
- E. stockOptions is not a member of class manager.

#### **QUESTION 47**

Given:

```
class Product {
    double price;
}

public class Test {
    public void updatePrice(Product product, double price) {
        price = price * 2;
        product.price = product.price + price;
    }
    public static void main(String[] args) {
        Product prt = new Product();
        prt.price = 200;
        double newPrice = 100;

        Test t = new Test();
        t.updatePrice(prt, newPrice);
        System.out.println(prt.price + " : " + newPrice);
    }
}
```

What is the result?

- A. 200.0 : 100.0
- B. 400.0 : 200.0
- C. 400.0 : 100.0
- D. Compilation fails.

**Answer: C**

**Explanation:**

After call to updatePrice prt.price change its value to 400 (prt is passed by reference)  
variable newPrice never changes its value from 100 (newPrice is passed by value)

#### QUESTION 48

Given the code fragment:

```
if (aVar++ < 10) {
    System.out.println(aVar + " Hello World!");
} else {
    System.out.println(aVar + " Hello Universe!");
}
```

What is the result if the integer aVar is 9?

- A. Hello World!
- B. Hello Universe!
- C. Hello World
- D. Compilation fails.

**Answer: A**

#### QUESTION 49

Given the code fragment:

```
public static void main(String[] args) {
    String date = LocalDate
        .parse("2014-05-04")
        .format(DateTimeFormatter.ISO_DATE_TIME);
    System.out.println(date);
}
```

What is the result?

- A. May 04, 2014T00:00:00.000
- B. 2014-05-04T00:00: 00. 000
- C. 5/4/14T00:00:00.000
- D. An exception is thrown at runtime.

**Answer:** D

**Explanation:**

The exception `java.time.temporal.UnsupportedTemporalTypeException` is thrown at runtime. We should use class `LocalDateTime` with `ISO_DATE_TIME` format or use the format `ISO_DATE` to avoid the exception.

See `ISO_DATE_TIME` at

<https://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html>

See examples at

<https://gist.github.com/mscharhag/9195718>

#### **QUESTION 50**

Given the code fragment:

```
public static void main(String[] args) {
    Short s1 = 200;
    Integer s2 = 400;
    Long s3 = (long) s1 + s2;           //line n1
    String s4 = (String) (s3 * s2);     //line n2
    System.out.println("Sum is " + s4);
}
```

What is the result?

- A. Sum is 600
- B. Compilation fails at line n1.
- C. Compilation fails at line n2.
- D. A `ClassCastException` is thrown at line n1.
- E. A `ClassCastException` is thrown at line n2.

**Answer:** C

**Explanation:**

Compilation fails at n2 because the compiler cannot cast long to String.

#### **QUESTION 51**

What is the name of the Java concept that uses access modifiers to protect variables and hide them within a class?

- A. Encapsulation

- B. Inheritance
- C. Abstraction
- D. Instantiation
- E. Polymorphism

**Answer:** A

**Explanation:**

Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

[http://www.tutorialspoint.com/java/java\\_access\\_modifiers.htm](http://www.tutorialspoint.com/java/java_access_modifiers.htm)

#### QUESTION 52

Given the code fragment:

```
abstract class Planet {  
    protected void revolve() { //line n1  
    }  
  
    abstract void rotate(); //line n2  
}  
  
class Earth extends Planet {  
    void revolve() { //line n3  
    }  
  
    protected void rotate() { //line n4  
    }  
}
```

Which two modifications, made independently, enable the code to compile?

- A. Make the method at line n1 public.
- B. Make the method at line n2 public.
- C. Make the method at line n3 public.
- D. Make the method at line n3 protected.
- E. Make the method at line n4 public.

**Answer:** CD

**Explanation:**

We can't assign weaker privileges in a subclass.

Method revolve() is declared protected in class Planet.

We can declare revolve() as public or protected in class Earth.

#### QUESTION 53

Given:

```
class Vehicle {  
    String type = "4W";  
    int maxSpeed = 100;  
  
    Vehicle(String type, int maxSpeed) {  
        this.type = type;  
        this.maxSpeed = maxSpeed;  
    }  
}  
  
class Car extends Vehicle {  
    String trans;  
  
    Car(String trans) {           //line n1  
        this.trans = trans;  
    }  
  
    Car(String type, int maxSpeed, String trans) {  
        super(type, maxSpeed);  
        this(trans);           //line n2  
    }  
}
```

And given the code fragment:

```
7. Car c1 = new Car("Auto");  
8. Car c2 = new Car("4W", 150, "Manual");  
9. System.out.println(c1.type + " " + c1.maxSpeed + " " + c1.trans);  
10. System.out.println(c2.type + " " + c2.maxSpeed + " " + c2.trans);
```

What is the result?

- A. 4W 100 Auto  
4W 150 Manual
- B. Null 0 Auto  
4W 150 Manual
- C. Compilation fails only at line n1
- D. Compilation fails only at line n2
- E. Compilation fails at both line n1 and line n2

**Answer: E**

**Explanation:**

Compilation fails at n1 because Vehicle hasn't a default constructor

Compilation fails at n2 because this() must be the first statement in constructor body

#### QUESTION 54

Given the code fragment:

```
1. class X {  
2.     public void printFileContent() {  
3.         /* code goes here */  
4.         throw new IOException();  
5.     }  
6. }  
7. public class Test {  
8.     public static void main(String[] args) {  
9.         X xobj = new X();  
10.        xobj.printFileContent();  
11.    }  
12. }
```

Which two modifications should you make so that the code compiles successfully?

- A) Replace line 8 with `public static void main(String[] args) throws Exception {`
  - B) Replace line 10 with:  
`try {  
 xobj.printFileContent();  
}  
catch(Exception e) { }  
catch(IOException e) { }`
  - C) Replace line 2 with `public void printFileContent() throws IOException {`
  - D) Replace line 4 with `throw IOException("Exception raised");`
  - E) At line 11, insert `throw new IOException();`
- 
- A. Option A
  - B. Option B
  - C. Option C
  - D. Option D
  - E. Option E

**Answer:** AC

**Explanation:**

Add throws clause in both `printFileContent` and `main`.

#### **QUESTION 55**

Given the following two classes:

```
public class Customer {
    ElectricAccount acct = new ElectricAccount();

    public void useElectricity(double kWh) {
        acct.addKWh(kWh);
    }
}

public class ElectricAccount {
    private double kWh;
    private double rate = 0.07;
    private double bill;

    //line n1
}
```

How should you write methods in the ElectricAccount class at line n1 so that the member variable bill is always equal to the value of the member variable kWh multiplied by the member variable rate? Any amount of electricity used by a customer (represented by an instance of the customer class) must contribute to the customer's bill (represented by the member variable bill) through the method useElectricity method.

An instance of the customer class should never be able to tamper with or decrease the value of the member variable bill.

C A) public void addKWh(double kWh) {  
    this.kWh += kWh;  
    this.bill = this.kWh\*this.rate;  
}  
  
C B) public void addKWh(double kWh) {  
    if (kWh > 0){  
        this.kWh += kWh;  
        this.bill = this.kWh \* this.rate;  
    }  
}  
  
C C) private void addKWh(double kWh) {  
    if (kWh > 0) {  
        this.kWh += kWh;  
        this.bill = this.kWh\*this.rate;  
    }  
}  
  
C D) public void addKWh(double kWh) {  
    if(kWh > 0) {  
        this.kWh += kWh;  
        setBill(this.kWh);  
    }  
}  
    public void setBill(double kWh) {  
        bill = kWh\*rate;  
    }  
}

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** D

**QUESTION 56**

Given the code fragments:

Person.java:

```
public class Person {  
    String name;  
    int age;  
  
    public Person(String n, int a) {  
        name = n;  
        age = a;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public int getAge() {  
        return age;  
    }  
}
```

Test.java:

```
public static void checkAge(List<Person> list, Predicate<Person> predicate) {  
    for (Person p : list) {  
        if (predicate.test(p)) {  
            System.out.println(p.name + " ");  
        }  
    }  
}  
  
public static void main(String[] args) {  
    List<Person> iList = Arrays.asList(new Person("Hank", 45),  
   new Person("Charlie", 40),  
   new Person("Smith", 38));  
    //line n1  
}
```

Which code fragment, when inserted at line n1, enables the code to print Hank?

- A. checkAge (iList, ( ) -> p. get Age ( ) > 40);
- B. checkAge(iList, Person p -> p.getAge( ) > 40);
- C. checkAge (iList, p -> p.getAge ( ) > 40);
- D. checkAge(iList, (Person p) -> { p.getAge() > 40; });

**Answer: C**

**Explanation:**

<https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html>

#### QUESTION 57

Given the code fragment:

```
public static void main(String[] args) {
    String[][] arr = {{"A", "B", "C"}, {"D", "E"}};
    for (int i = 0; i < arr.length; i++) {
        for (int j = 0; j < arr[i].length; j++) {
            System.out.print(arr[i][j] + " ");
            if (arr[i][j].equals("B")) {
                break;
            }
        }
        continue;
    }
}
```

What is the result?

- A. A B C
- B. A B C D E
- C. A B D E
- D. Compilation fails.

**Answer:** C

**QUESTION 58**

Given the code fragment:

```
public static void main(String[] args) {
    String str = " ";
    str.trim();
    System.out.println(str.equals("") + " " + str.isEmpty());
}
```

What is the result?

- A. true true
- B. true false
- C. false false
- D. false true

**Answer:** C

**QUESTION 59**

Given:

```
class CD {  
    int r;  
    CD(int r) {  
        this.r=r;  
    }  
}  
  
class DVD extends CD {  
    int c;  
    DVD(int r, int c) {  
        // line n1  
    }  
}
```

And given the code fragment:

```
DVD dvd = new DVD(10,20);
```

Which code fragment should you use at line n1 to instantiate the dvd object successfully?

- A) super.r = r;  
 this.c = c;
- B) super(r);  
 this(c);
- C) super(r);  
 this.c = c;
- D) this.c = r;  
 super(c);

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer: C**

#### QUESTION 60

Given the code fragment:

```
int a[] = {1, 2, 3, 4, 5};  
for(XXX) {  
    System.out.print(a[e]);  
}
```

Which option can replace xxx to enable the code to print 135?

- A. int e = 0; e <= 4; e++
- B. int e = 0; e < 5; e += 2
- C. int e = 1; e <= 5; e += 1



Leader of IT Certifications

[Free VCE and PDF Exam Dumps from PassLeader](#)

D. int e = 1; e < 5; e+ =2

**Answer:** B

**Explanation:**

This loop prints the array elements with index 0, 2 and 4.

These elements are 1, 3, 5.

[Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps](#)

➤ **Vendor: Oracle**

➤ **Exam Code: 1Z0-808**

➤ **Exam Name: Java SE 8 Programmer I**

➤ **Question 61 -- Question 80**

[\*\*Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps\*\*](#)

**QUESTION 61**

Which statement best describes encapsulation?

- A. Encapsulation ensures that classes can be designed so that only certain fields and methods of an object are accessible from other objects.
- B. Encapsulation ensures that classes can be designed so that their methods are inheritable.
- C. Encapsulation ensures that classes can be designed with some fields and methods declared as abstract.
- D. Encapsulation ensures that classes can be designed so that if a method has an argument MyType x, any subclass of MyType can be passed to that method.

**Answer:** A

**Explanation:**

[http://www.tutorialspoint.com/java/java\\_encapsulation.htm](http://www.tutorialspoint.com/java/java_encapsulation.htm)

**QUESTION 62**

Given the code fragment from three files:

SalesMan.java:

```
package sales;
public class SalesMan { }
```

Product.java:

```
package sales.products;
public class Product { }
```

Market.java:

```
1. package market;
2. // insert code here
3. public class USMarket {
4.     SalesMan sm;
5.     Product p;
6. }
```

Which code fragment, when inserted at line 2, enables the code to compile?

- A) import sales.\*;
- B) import java.sales.products.\*;
- C) import sales;  
import sales.products;
- D) import sales.\*;  
import products.\*;
- E) import sales.\*;  
import sales.products.\*;

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer: E**

**Explanation:**

<https://docs.oracle.com/javase/tutorial/java/package/usepkgs.html>

### QUESTION 63

Given the following class:

```

public class CheckingAccount {
    public int amount;
    public CheckingAccount(int amount) {
        this.amount = amount;
    }
    public int getAmount() {
        return amount;
    }
    public void changeAmount(int x) {
        amount += x;
    }
}

```

And given the following main method, located in another class:

```

public static void main(String[] args) {
    CheckingAccount acct = new CheckingAccount((int)(Math.random()*1000));
    //line n1
    System.out.println(acct.getAmount());
}

```

Which three lines, when inserted independently at line n1, cause the program to print a balance?

- A. this.amount = 0;
- B. amount = 0;
- C. acct(0);
- D. acct.amount = 0;
- E. acct.getAmount() = 0;
- F. acct.changeAmount(0);
- G. acct.changeAmount(-acct.amount);
- H. acct.changeAmount(-acct.getAmount());

**Answer: DGH**

**Explanation:**

A and B don't compile because there isn't a variable amount in method main.

C is wrong because we can't call the constructor acct directly.

E is wrong because we can't make a method on acct equal to 0.

F is wrong because does not change variable amount of class CheckingAccount.

#### QUESTION 64

Given the code fragment:

```

String shirts[][] = new String[2][2];
shirts[0][0] = "red";
shirts[0][1] = "blue";
shirts[1][0] = "small";
shirts[1][1] = "medium";

```

Which code fragment prints red: blue: small: medium?

C A) for (int index = 1; index < 2; index++) {  
    for (int idx = 1; idx < 2; idx++) {  
        System.out.print(shirts[index][idx] + ":" );  
    }  
}  
  
C B) for (int index = 0; index < 2; ++index) {  
    for (int idx = 0; idx < index; ++idx) {  
        System.out.print(shirts[index][idx] + ":" );  
    }  
}  
  
C C) for (String c : colors) {  
    for (String s : sizes) {  
        System.out.println(s + ":" );  
    }  
}  
  
C D) for (int index = 0; index < 2;) {  
    for (int idx = 0; idx < 2;) {  
        System.out.print(shirts[index][idx] + ":" );  
        idx++;  
    }  
    index++;  
}

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** B

**QUESTION 65**

Given the code fragment:

```
public class Test{  
  
    void readCard(int cardNo) throws Exception {  
        System.out.println("Reading Card");  
    }  
  
    void checkCard(int cardNo) throws RuntimeException { // line n1  
        System.out.println("Checking Card");  
    }  
  
    public static void main(String[] args) {  
        Test ex = new Test();  
        int cardNo = 12344;  
        ex.checkCard(cardNo);  
        ex.readCard(cardNo);  
    }  
}
```

What is the result?

- A. Reading Card  
Checking Card
- B. Compilation fails only at line n1.
- C. Compilation fails only at line n2.
- D. Compilation fails only at line n3.
- E. Compilation fails at both line n2 and line n3.

**Answer:** D

**Explanation:**

Exception is a checked exception so we are required to check it with try/catch or be declared in method main.

#### QUESTION 66

Given the code fragment:

```
public static void main(String[] args) {  
    StringBuiler sb = new StringBuiler(5);  
    String s = "";  
  
    if (sb.equals(s)) {  
        System.out.println("Match 1");  
    } else if (sb.toString().equals(s.toString())) {  
        System.out.println("Match 2");  
    } else {  
        System.out.println("No Match");  
    }  
}
```

What is the result?

- A. Match 1
- B. Match 2
- C. No Match

- D. A NullPointerException is thrown at runtime.

**Answer:** B

**QUESTION 67**

Given:

```
package p1;
public class Acc {
    int p;
    private int q;
    protected int r;
    public int s;
}
```

Test.java:

```
package p2;
import p1.Acc;
public class Test extends Acc {
    public static void main(String[] args) {
        Acc obj = new Test();
    }
}
```

Which statement is true?

- A. Both p and s are accessible by obj.
- B. Only s is accessible by obj.
- C. Both r and s are accessible by obj.
- D. p, r, and s are accessible by obj.

**Answer:** B

**Explanation:**

Only s is accessible because it is the only public member of class Acc.

**QUESTION 68**

Given:

Base.java:

```
class Base {  
    public void test(){  
        System.out.println("Base ");  
    }  
}
```

DerivedA.java:

```
class DerivedA extends Base {  
    public void test(){  
        System.out.println("DerivedA ");  
    }  
}
```

DerivedB.java:

```
class DerivedB extends DerivedA {  
    public void test(){  
        System.out.println("DerivedB ");  
    }  
    public static void main(String[] args) {  
        Base b1 = new DerivedB();  
        Base b2 = new DerivedA();  
        Base b3 = new DerivedB();  
        b1 = (Base) b3;  
        Base b4 = (DerivedA) b3;  
        b1.test();  
        b4.test();  
    }  
}
```

What is the result?

- A. Base  
DerivedA
- B. Base  
DerivedB
- C. DerivedB  
DerivedB
- D. DerivedB  
DerivedA
- E. A classcast Except ion is thrown at runtime.

**Answer: C**

#### QUESTION 69

Given the code fragment:

```
public static void main(String[] args) {
    ArrayList myList = new ArrayList();
    String[] myArray;
    try {
        while (true) {
            myList.add("My String");
        }
    } catch (RuntimeException re) {
        System.out.println("Caught a RuntimeException");
    } catch (Exception e) {
        System.out.println("Caught an Exception");
    }
    System.out.println("Ready to use");
}
```

What is the result?

- A. Execution terminates in the first catch statement, and caught a RuntimeException is printed to the console.
- B. Execution terminates in the second catch statement, and caught an Exception is printed to the console.
- C. A runtime error is thrown in the thread "main".
- D. Execution completes normally, and Ready to use is printed to the console.
- E. The code fails to compile because a throws keyword is required.

**Answer: C**

**Explanation:**

while loop is an infinite loop so the program ends with an OutOfMemoryError.

This error can't be caught with Exception nor RuntimeException.

<http://stackoverflow.com/questions/1692230/is-it-possible-to-catch-out-of-memory-exception-in-java>

### QUESTION 70

Given:

```
System.out.println("5 + 2 = " + 3 + 4);
System.out.println("5 + 2 = " + (3 + 4));
```

What is the result?

- A) 5 + 2 = 34  
5 + 2 = 34
- B) 5 + 2 + 3 + 4  
5 + 2 = 7
- C) 7 = 7  
7 + 7
- D) 5 + 2 = 34  
5 + 2 = 7

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** D

**Explanation:**

If neither operand of + is a reference to a String object, the operator is the arithmetic addition operator, not the string concatenation operator. Note that Java does not allow a program to define overloaded operators. However, the language defines the + operator to have a meaning that is fundamentally different from arithmetic addition if at least one of its operands is a String object.

The way in which Java decides if + means arithmetic addition or string concatenation means that the use of parentheses can alter the meaning of the + operator.

See “String Concatenation Operator +” at

[http://oponet.stsci.edu/web/documentation/Java%20Reference%20Library%201.02/langref/ch04\\_06.htm](http://oponet.stsci.edu/web/documentation/Java%20Reference%20Library%201.02/langref/ch04_06.htm)

### QUESTION 71

Given:

```
public static void main(String[] args) {  
    String ta = "A ";  
    ta = ta.concat("B ");  
    String tb = "C ";  
    ta = ta.concat(tb);  
    ta.replace('C', 'D');  
    ta = ta.concat(tb);  
    System.out.println(ta);  
}
```

What is the result?

- A. A B C D
- B. A C D
- C. A B C C
- D. A B D
- E. A B D C

**Answer:** C

**Explanation:**

The line “ta.replace('C', 'D');” returns a string that is never assigned to ta.

### QUESTION 72

Given the code fragment:

```
3. public static void main(String[] args) {
4.     int x = 5;
5.     while (isAvailable(x)) {
6.         System.out.print(x);
7.
8.     }
9. }
10.
11. public static boolean isAvailable(int x) {
12.     return x-- > 0 ? true : false;
13. }
```

Which modification enables the code to print 54321?

- A. Replace line 6 with System.out.print(--x);
- B. At line 1, insert x --;
- C. Replace line 6 with --x; and, at line 7, insert system.out.print(x);
- D. Replace line 12 With return (x > 0) ? false: true;

**Answer:** A

### QUESTION 73

Given the code fragment:

```
4. public static void main(String[] args) {
5.     boolean opt = true;
6.     switch (opt) {
7.         case true:
8.             System.out.print("True");
9.             break;
10.        default:
11.            System.out.print("****");
12.        }
13.        System.out.println("Done");
14. }
```

Which modification enables the code fragment to print TrueDone?

- A. Replace line 5 With String result = "true";  
Replace line 7 with case "true":
- B. Replace line 5 with boolean opt = l;  
Replace line 7 with case 1=
- C. At line 9, remove the break statement.
- D. Remove the default section.

**Answer:** A

**Explanation:**

Switch statements with String cases were implemented in Java SE 7.

### QUESTION 74

Given the following main method:

```
public static void main(String[] args) {
    int num = 5;
    do {
        System.out.print(num-- + " ");
    } while(num == 0);
}
```

What is the result?

- A. 5 4 3 2 1 0
- B. 5 4 3 2 1
- C. 4 2 1
- D. 5
- E. Nothing is printed

**Answer:** D

**Explanation:**

The loop body executes only once because on the while condition num = 4.

When the execution reaches System.out.print, num = 5.

### QUESTION 75

Given the code fragment:

```
int x = 100;
int a = x++;
int b = ++x;
int c = x++;
int d = (a < b) ? (a < c) ? a: (b < c )? b: c;
System.out.println(d);
```

What is the result?

- A. 100
- B. 101
- C. 102
- D. 103
- E. Compilation fails

**Answer:** E

**Explanation:**

Compilation fails with error ": expected" because we have three ternary operators but only two colons.

### QUESTION 76

Given:

```
public class Test {

    public static void main(String[] args) {

        String[][] chs = new String[2][];
        chs[0] = new String[2];
        chs[1] = new String[5];
        int i = 97;

        for (int a = 0; a < chs.length; a++) {
            for (int b = 0; b < chs.length; b++) {
                chs[a][b] = "" + i;
                i++;
            }
        }

        for (String[] ca : chs) {
            for (String c : ca) {
                System.out.print(c + " ");
            }
            System.out.println();
        }
    }
}
```

What is the result?

- A. 97 98  
99 100 null null null
- B. 91 98  
99 100 101 102 103
- C. Compilation rails.
- D. A NullPointerException is thrown at runtime.
- E. An ArrayIndexOutOfBoundsException is thrown at runtime.

**Answer: A**

**Explanation:**

When we exit first loop we have

chs[0][0] = 97

chs[0][1] = 98

chs[1][0] = 99

chs[1][1] = 100

chs[1][2] = null;

chs[1][3] = null;

chs[1][4] = null;

The second loop prints these values.

### QUESTION 77

Given the code fragment:

```
public class Employee {  
    String name;  
    boolean contract;  
    double salary;  
    Employee() {  
        // line n1  
    }  
    public String toString(){  
        return name + ":" + contract + ":" + salary;  
    }  
    public static void main(String[] args) {  
        Employee e = new Employee();  
        // line n2  
        System.out.print(e);  
    }  
}
```

Which two modifications, when made independently, enable the code to print joe:true: 100.0?

- A) Replace line n2 with:

```
e.name = "Joe";  
e.contract = true;  
e.salary = 100;
```

- B) Replace line n2 with:

```
this.name = "Joe";  
this.contract = true;  
this.salary = 100;
```

- C) Replace line n1 with:

```
this.name = new String("Joe");  
this.contract = new Boolean(true);  
this.salary = new Double(100);
```

- D) Replace line n1 with:

```
name = "Joe";  
contract = TRUE;  
salary = 100.0f;
```

- E) Replace line n1 with:

```
this("Joe", true, 100);
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D
- E. Option E

**Answer:** AC**QUESTION 78**

View the exhibit:

```
public class Student {  
    public String name = "";  
    public int age = 0;  
    public String major = "Undeclared";  
    public boolean fulltime = true;  
    public void display() {  
        System.out.println("Name: " + name + " Major: " + major); }  
    public boolean isFullTime() {  
        return fulltime;  
    }  
}
```

Which line of code initializes a student instance?

- A. Student student1;
- B. Student student1 = Student.new();
- C. Student student1 = new Student();
- D. Student student1 = Student();

**Answer:** C**QUESTION 79**

What should keyword1 and keyword2 be respectively, in order to produce output 2345?

```
int [] array = {1,2,3,4,5};  
for (int i: array) {  
    if ( i < 2) {  
        keyword1 ;  
    }  
    System.out.println(i);  
    if ( i == 3) {  
        keyword2 ;  
    } }
```

- A. continue, break
- B. break, break
- C. break, continue
- D. continue, continue

**Answer:** D**QUESTION 80**

What is the result?

```
int i, j=0;  
i = (3* 2 +4 +5 ) ;  
j = (3 * ((2+4) + 5));  
System.out.println("i:"+ i + "\nj":+j);
```



Leader of IT Certifications

[Free VCE and PDF Exam Dumps from PassLeader](#)

- A. i: 16  
j: 33
- B. i: 15  
j: 33
- C. i: 33  
j: 23
- D. i: 15  
j: 23

- A. Option A
- B. Option B
- C. Option A
- D. Option D

**Answer:** B

[Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps](#)

➤ **Vendor: Oracle**

➤ **Exam Code: 1Z0-808**

➤ **Exam Name: Java SE 8 Programmer I**

➤ **Question 81 -- End**

[Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps](#)

**QUESTION 81**

What is the result?

```
boolean log3 = ( 5.0 != 6.0) && ( 4 != 5);  
boolean log4 = (4 != 4) || (4 == 4);  
System.out.println("log3:" + log3 + \nlog4" + log4);
```

- A. log3:false  
log4:true
- B. log3:true  
log4:true
- C. log3:true  
log4:false
- D. log3:false  
log4:false

**Answer:** B

**QUESTION 82**

Which statement will empty the contents of a StringBuilder variable named sb?

- A. sb.deleteAll();
- B. sb.delete(0, sb.size());
- C. sb.delete(0, sb.length());
- D. sb.removeAll();

**Answer:** C

**QUESTION 83**

What is the result?

```
Class StaticField {  
    static int i = 7;  
    public static void main(String[] args) {  
        StaticFied obj = new StaticField();  
        obj.i++;  
        StaticField.i++;  
        obj.i++;  
        System.out.println(StaticField.i + " " + obj.i);  
    }  
}
```

- A. 10 10
- B. 8 9
- C. 9 8
- D. 7 10

**Answer:** A

#### QUESTION 84

Which two are valid array declaration?

- A. Object array[];
- B. Boolean array[3];
- C. int[] array;
- D. Float[2] array;

**Answer:** AC

#### QUESTION 85

Given:

```
class Overloading {  
    int x(double d) {  
        System.out.println("one");  
        return 0;  
    }  
    String x(double d) {  
        System.out.println("two");  
        return null;  
    }  
    double x(double d) {  
        System.out.println("three");  
        return 0.0;  
    }  
    public static void main(String[] args) {  
        new Overloading().x(4.0);  
    }  
}
```

What is the result?

- A. one
- B. two
- C. three
- D. Compilation fails.

**Answer:** D

### QUESTION 86

Given:

```
public class MainMethod {  
    void main() {  
        System.out.println("one");  
    }  
    static void main(String args) {  
        System.out.println("two");  
    }  
    public static void main(String[] args) {  
        System.out.println("three");  
    }  
    void mina(Object[] args) {  
        System.out.println("four");  
    }  
}
```

What is printed out when the program is executed?

- A. one
- B. two
- C. three
- D. four

**Answer:** C

### QUESTION 87

Given:

```
public class ScopeTest {  
    int j, int k;  
    public static void main(String[] args) {  
        new ScopeTest().doStuff();  
    }  
    void doStuff() {  
        int x = 5;  
        doStuff2();  
        System.out.println("x");  
    }  
    void doStuff2() {  
        int y = 7;  
        System.out.println("y");  
        for (int z = 0; z < 5; z++) {  
            System.out.println("z");  
            System.out.println("y");  
        }  
    }  
}
```

Which two items are fields?

- A. j
- B. k
- C. x
- D. y
- E. z

**Answer:** AB

**QUESTION 88**

A method is declared to take three arguments.  
A program calls this method and passes only two arguments.  
What is the results?

- A. Compilation fails.
- B. The third argument is given the value null.
- C. The third argument is given the value void.
- D. The third argument is given the value zero.
- E. The third argument is given the appropriate falsy value for its declared type.
- F. An exception occurs when the method attempts to access the third argument.

**Answer:** A

**QUESTION 89**

Which three are valid replacements for foo so that the program will compiled and run?

```
public class ForTest {  
    public static void main(String[] args) {  
        int[] arrar = {1,2,3};  
        for ( foo ) {  
        }  
    }  
}
```

- A. int i: array
- B. int i = 0; i < 1; i++
- C. ;;
- D. ; i < 1; i++
- E. ; i < 1;

**Answer:** ABC

**QUESTION 90**

Given:

```
public class SampleClass {  
    public static void main(String[] args) {  
        AnotherSampleClass asc = new AnotherSampleClass(); SampleClass sc = new  
        SampleClass();  
        sc = asc;  
        System.out.println("sc: " + sc.getClass());  
        System.out.println("asc: " + asc.getClass());  
    }  
}  
class AnotherSampleClass extends SampleClass {  
}
```

What is the result?

- A. sc: class Object  
asc: class AnotherSampleClass
- B. sc: class SampleClass  
asc: class AnotherSampleClass
- C. sc: class AnotherSampleClass

- asc: class SampleClass
- D. sc: class AnotherSampleClass  
asc: class AnotherSampleClass

**Answer:** D

### QUESTION 91

Given the code fragment:

```
int b = 3;
if ( !(b > 3) ) {
    System.out.println("square");
}
System.out.println("circle");
System.out.println("...");
```

What is the result?

- A. square...
- B. circle...
- C. squarecircle...
- D. Compilation fails.

**Answer:** C

### QUESTION 92

What is the proper way to define a method that takes two int values and returns their sum as an int value?

- A. int sum(int first, int second) { first + second; }
- B. int sum(int first, second) { return first + second; }
- C. sum(int first, int second) { return first + second; }
- D. int sum(int first, int second) { return first + second; }
- E. void sum (int first, int second) { return first + second; }

**Answer:** D

#### Explanation:

Incorrect answers:

A: no return statement

### QUESTION 93

Which two are Java Exception classes?

- A. SecurityException
- B. DuplicatePathException
- C. IllegalArgumentException
- D. TooManyArgumentsException

**Answer:** AC

### QUESTION 94

Given the for loop construct:

```
for ( expr1 ; expr2 ; expr3 ) {
```

```
statement;  
}
```

Which two statements are true?

- A. This is not the only valid for loop construct; there exists another form of for loop constructor.
- B. The expression expr1 is optional.  
it initializes the loop and is evaluated once, as the loop begins.
- C. When expr2 evaluates to false, the loop terminates.  
It is evaluated only after each iteration through the loop.
- D. The expression expr3 must be present.  
It is evaluated after each iteration through the loop.

**Answer:** AB

**Explanation:**

A is true because there are two types of for loop in Java. Classic and Enhanced.

B is true because we can run code like for( ; expr2 ; expr3).

C is false because expr2 is evaluated BEFORE each iteration.

D is false because we can run code like for(expr1; expr2; ).

<http://www.java-tips.org/java-se-tips-100019/24-java-lang/480-the-enhanced-for-loop.html>

### QUESTION 95

What is the result?

```
public class StringReplace {  
    public static void main(String[] args) {  
        String message = "Hi everyone!";  
        System.out.println("message = " + message.replace("e", "X"));  
    }  
}
```

- A. message = Hi everyone!
- B. message = Hi XvXryonX!
- C. A compile time error is produced.
- D. A runtime error is produced.
- E. message =
- F. message = Hi Xeveryone!

**Answer:** B

### QUESTION 96

Which two statements are true for a two-dimensional array?

- A. It is implemented as an array of the specified element type.
- B. Using a row by column convention, each row of a two-dimensional array must be of the same size
- C. At declaration time, the number of elements of the array in each dimension must be specified
- D. All methods of the class Object may be invoked on the two-dimensional array.

**Answer:** AD

### QUESTION 97

Which three statements are benefits of encapsulation?

- A. allows a class implementation to change without changing the clients

- B. protects confidential data from leaking out of the objects
- C. prevents code from causing exceptions
- D. enables the class implementation to protect its invariants
- E. permits classes to be combined into the same package
- F. enables multiple instances of the same class to be created safely

**Answer:** ABD

### QUESTION 98

Given the code fragment:

1. `ArrayList<Integer> list = new ArrayList<>(1);`
2. `list.add(1001);`
3. `list.add(1002);`
4. `System.out.println(list.get(list.size()));`

What is the result?

- A. Compilation fails due to an error on line 1.
- B. An exception is thrown at run time due to error on line 3
- C. An exception is thrown at run time due to error on line 4
- D. 1002

**Answer:** C

#### Explanation:

The code compiles fine.

At runtime an `IndexOutOfBoundsException` is thrown when the second list item is added.

### QUESTION 99

Given the code fragment:

```
String[] colors = {"red", "blue", "green", "yellow", "maroon", "cyan"};
```

Which code fragment prints blue, cyan, ?

```
C A) for (String c:colors){  
    if (c.length() != 4) {  
        continue;  
    }  
    System.out.print(c+", ");  
}  
  
C B) for (String c:colors[]) {  
    if (c.length() <= 4) {  
        continue;  
    }  
    System.out.print(c+", ");  
}  
  
C C) for (String c:String[] colors) {  
    if (c.length() >= 3) {  
        continue;  
    }  
    System.out.print(c+", ");  
}  
  
C D) for (String c:colors){  
    if (c.length() != 4) {  
        System.out.print(c+", ");  
        continue;  
    }  
}
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer:** A

#### **QUESTION 100**

View the Exhibit.

```
public class Hat {  
public int ID =0;  
public String name = "hat";  
public String size = "One Size Fit All";  
public String color="";  
public String getName() { return name; }  
public void setName(String name) {  
this.name = name;  
}  
}
```

Given:

```
public class TestHat {
```

```
public static void main(String[] args) {  
    Hat blackCowboyHat = new Hat();  
}  
}
```

Which statement sets the name of the Hat instance?

- A. blackCowboyHat.setName = "Cowboy Hat";
- B. setName("Cowboy Hat");
- C. Hat.setName("Cowboy Hat");
- D. blackCowboyHat.setName("Cowboy Hat");

**Answer:** D

**QUESTION 101**

Which code fragment cause a compilation error?

- A. float flt = 100F;
- B. float flt = (float) 1\_11.00;
- C. float flt = 100;
- D. double y1 = 203.22;  
 float flt = y1;
- E. int y2 = 100;  
 float flt = (float) y2;

**Answer:** .....

[\*\*Visit PassLeader and Download Full Version 1Z0-808 Exam Dumps\*\*](#)