

Oracle 1z0-808



Java SE 8 Programmer I

Version: 5.4

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

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: D

Explanation:

QUESTION NO: 8

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: 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 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

**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: C**

**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); //line n2
 ex.readCard(cardNo); //line n3
 }
}
```

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. }
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);

--x will be replaced by 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**

**Explanation:**

There is no line 1 in the exhibit..!!!!

**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 result = "true";

Replace line 7 with case "true":

**B.**

Replace line 5 with boolean opt = 1;

Replace line 7 with case 1=

**C.**

At line 9, remove the break statement.

**D.**

Remove the default section.

**Answer: C**

**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[a].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 fails.

**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:**

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

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- none of the above  
int [ ] array= new int[2]/  
OR  
int array[ ]=new int[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: C**

**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: D**

**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:



```
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 Except ion 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:**

NOTE: I am not sure at all about the answers.

**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();
 setName(name);
 }

 public Person(String name, int age) {
 Person(name);
 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 pl;
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:

Process all the elements of the array in the order of entry.

Process all the elements of the array in the reverse order of entry.

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.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**

**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**