

1

Consider the following code segment:

```
for (int i = 0; i < 5; i++){  
    for (int j = 9; j >= 1; j--){  
        System.out.println("!");  
    }  
}
```

How many times will a '!' be printed?

- 35
- 40
- 38
- 42
- 45

2

```
public class Shape {  
    public void what() { System.out.print("Shape ");}  
}  
  
public class Rectangle extends Shape {  
    public void what() { System.out.print("Rectangle "); }  
}  
  
public class Square extends Rectangle { }  
  
    public class Oval extends Shape {  
        public void what() { System.out.print("Oval "); }  
    }  
  
    public class Circle extends Oval { }
```

What is the output from looping through the following array and calling what on each?

```
Shape[] shapes = {new Shape(), new Rectangle(), new Square(), new  
Circle()};
```

- Shape Rectangle Rectangle Circle
- Shape Shape Shape Shape
- Shape Rectangle Square Circle
- Shape Rectangle Rectangle Oval
- There will be a compile time error

3

Given the following code:

```
String s1 = new String("hi");  
String s2 = new String("hi there");  
String s3 = s2.substring(0,1);  
String s4 = new String("hi");
```

Which of the following would return true?

- I. `s1.equals(s3)`
- II. `s1 == s4`
- III. `s1.equals(s4)`

- II only
- I and III only
- III only
- I only
- II and III only

4

When is the following Boolean expression true (a and b are integers)?

`(a < b) && !(b < a)`

- (A) (B) Never true
- (C) (D) (E)

- Always true
- `a > b`
- `a < b`
- `a = b`
- Never true

5

Given the following class declarations:

```
public class Car {
    private String make;

    public Car () {
        make = "Ford"; }

    public Car(String theMake) {
        make = theMake; }

    public String getMake() {
        return make;
    }
}

public class ElectricCar extends Car {
    public ElectricCar(String theMake) {
        super(theMake); }
}
```

Which of the following will cause a compile time error?

- Car myCar = new ElectricCar();
- ElectricCar myCar = new ElectricCar("Ford");
- Car myCar = new ElectricCar("Toyota");
- Car myCar = new Car("Ford");
- Car myCar = new Car();

6

Consider the following code segment.

```
ArrayList<String> list1 = new ArrayList<String>();  
list1.add("a");  
list1.add("b");  
list1.add(0, "c");  
list1.add(1, "d");  
list1.set(2, "e");  
list1.add("f");  
System.out.println(list1);
```

What is printed as a result of executing the code segment?

- [c, d, e, b, f]
- [c, a, e, b, f]
- [c, d, e, a, b, f]
- [c, d, e, f]
- [c, a, e, d, b, f]

Given the following code:

```
public static int mystery(String str)
{
    if (str.length() == 1) return 0;
    else
    {
        if (str.substring(0,1).equals("y")) return 1 +
            mystery(str.substring(1));
        else return mystery(str.substring(1));
    }
}
```

What will it return when called with `mystery("xyyxyxy")`?

- 4
- 1
- 7
- 0
- 3

Given the following class definition. What are the values of x, y, and z just before method2 returns?

```
public class Class1
{
    public static int method1(int a, int b)
    {
        int temp = a;
        a = b;
        b = temp;
        return b;
    }

    public static void method2()
    {
        int x = 3;
        int y = 5;
        int z = method1(x, y);
    }
}
```

- x=3, y=3, z=3
- x=3, y=5, z=3
- x=5, y=3, z=5
- x=5, y=3, z=3
- x=3, y=5, z=5

Given the following class declarations.

```
public class Vehicle {
    // constructors not shown
    public void go()
    { // code not shown}
}

public class Hybrid extends Vehicle {
    // constructors not shown
    public void switchToElectric()
    { // code not shown }
}
```

Assume that the following declaration is in a different class.
Vehicle v = new Hybrid();

Which of the following will compile without error?

- I. v.go();
- II. v.switchToElectric();
- III. ((Hybrid) v).switchToElectric();

- I only
- I, II, and III
- II and III only
- I and III only
- II only

10

Consider the following code segment:

```
public static boolean check(String s)
{
    return s.length() >= 2 &&
        (s.substring(0,1).equals(s.substring(1,2)) ||
        check(s.substring(1)));
}
```

Pick the answer below that best describes all the cases when this method will return true.

- s starts with two or more of the same characters
- s contains only two characters
- s contains two or more of the same characters
- s ends with two or more of the same characters
- s contains two or more of the same characters in a row

11

Consider the following partial class definitions.

```
public class Rodent { // methods not shown}
public class Rat extends Rodent { // methods not shown}
public class Mouse extends Rodent { // more info }
public class LabRat extends Rat { // methods not shown}
```

Which of the following declarations would *not* cause a compile time error?

(A) (B) (C) (D)

- Rat rat = new Mouse();
- LabRat labRat = new Rat();
- Mouse mouse = new Rat();
- Rodent rodent = new Rat();
- Rat rat = new Rodent();

12

Which of the following correctly shows the iterations of an ascending (from left to right) insertion sort on an array with the following elements: {7,3,8,5,2}?

- {3,7,8,5,2}, {3,5,7,8,2}, {2,3,5,7,8}
- {2,3,8,5,7}, {2,3,8,5,7}, {2,3,5,8,7}, {2,3,5,7,8}
- {2,7,3,8,5}, {2,3,7,8,5}, {2,3,5,7,8}
- {2,3,8,5,7}, {2,3,5,8,7}, {2,3,5,7,8}
- {3,7,8,5,2}, {3,7,8,5,2}, {3,5,7,8,2}, {2,3,5,7,8}

13

Consider the following code segment:

```
int p = 5;
int q = 2;
int sum = 0;

while (p <= 7)
{
    sum += p % q;
    p++;
    q++;
}
```

What is the value of sum after the code is executed?

- 7
- 2
- 1
- 4
- 3

Consider the following method. What is the output from `conditionTest(3,-2)`?

```
public static void conditionTest(int num1, int num2)
{
    if ((num1>0) && (num2>0)) {
        if (num1>num2)
            System.out.println("A");
        else
            System.out.println("B");
    }
    else if ((num2<0) || (num1<0)) {
        System.out.println("C");
    }
    else if (num2 < 0) {
        System.out.println("D");
    }
    else {
        System.out.println("E");
    }
}
```

E

B

C

A

D

Consider the following code segment

```
public static void test(String str, int y)
{
    str = str + "bow";
    y = y * 2;
}
```

What are the values of `s` and `b` after the following has executed?

```
String s = "rain";
int b = 4;
test(s,b);
```

(A) (B)
(C) (D) (E)

- s="rain"; b=8
- s="rainbow"; b=4;
- s="bow"; b=4;
- s="rainbow"; b=8;
- s="rain"; b=4;

Consider the following code segment.

```
String str = "mnopqr";  
for (int x = 0; x < str.length() - 1; x++)  
{  
    System.out.print(str.substring(x, x + 2));  
}
```

What is printed as a result of executing the code segment?

- mnopqr
- Nothing is printed because an `IndexOutOfBoundsException` is thrown
- mnnooppqqr
- mnonopopqqr
- mmnnooppqrr

Given that `count` and `n` are both integer values, which of the following is true?

```
// Code block I
for (count = 0; count <= n; count++) {
    System.out.println(count);
}
```

```
//Code block II
count = 0;
while (count <= n) {
    count = count + 1;
    System.out.println(count);
}
```

- The output from I and II is only the same when `n=0`
- The output from I and II is never the same
- The output from I and II is the same for even values of `n`.
- The output from I and II is the same for all values of `n` except when `n=0`
- The output from I and II is the same for all values of `n`.

Given the following values for a 2D array `m` and the following code

1	1	1	1
1	2	3	4
2	2	2	2
2	4	6	8

```
int sum = 0;
for (int k = 0; k < m.length; k++) {
    sum = sum + m[m.length-1-k][1];
}
```

What is the value of `sum` after this code executes?

- 6
- 20
- 9
- 10
- 4

Consider the following method.

```
public static boolean test(int[] a, int val) {
    boolean temp = false;
    for ( int i = a.length-1; i >= 0; i--) {
        temp = ( a[i] == val );
        if (temp)
            return temp;
    }
    return temp;
}
```

Which of the below best describes all the cases in which `test` returns true?

- Whenever the last element in `a` is equal to `val`.
- Whenever the first element in `a` is equal to `val`.
- Whenever more than one element in `a` is equal to `val`.
- Whenever `a` contains any element which equals `val`.
- Whenever exactly 1 element in `a` is equal to `val`.

20

Consider the following method:

```
public static int m1(int a) {  
    if (a == 1)  
        return 1;  
    else  
        return 10 + m1(a - 1);  
}
```

What is the output when m1(6) is called?

- 60
- 51
- 11
- 61
- 1

Consider the following class definitions.

```
public class Student {
    public String getFood() {
        return "Pizza";
    }
    public String getInfo() {
        return "Studying";
    }
}
public class GradStudent extends Student {
    public String getFood() {
        return "Taco";
    }
    public String getInfo() {
        super.getInfo();
        return "Eating";
    }
}
```

What is printed when the following code is executed?

```
Student s = new GradStudent();
```

```
System.out.println(s.getInfo());
```

- Studying
Eating
- Studying
- Taco
- Pizza
- Eating

22

What are the values of a and b after the following loop finishes?

```
int a = 8, b = 3, temp = 0;
for (int i = 1; i < 6; i++) {
    temp = a;
    a = i + b;
    b = temp - i;
}
```

- a=9, b=3
- a=6, b=5
- a=0, b=13
- a=13, b=0
- a=11, b=0

23

Which of the following statements about interfaces is (are) true?

- I. One interface can inherit from another
- II. All methods declared in an interface are abstract methods (can't have a method body).
- III. All methods declared in an interface are public methods.

- I only
- I, II, and III only
- I and II only
- II only
- III only

Consider the following partial class definitions.

```
public class Person implements Comparable<Person>
{
    public int compareTo(Person other)
    {
        // code not shown
        return 0;
    }
}
public class Player extends Person
{ // code for class
}
```

Which of the following will cause a compile time error?

- Person p = new Person();
- Person p = new Player();
- Comparable p = new Person();
- Comparable p = new Player();
- Player p = new Person();

25

What is the value of `x` after the following code executes?

```
int x = 2;
int y = -1;
while (x < 30) {
    x = (int) Math.pow(x,
        Math.abs(y * 2));
}
```

- 16
- 32
- 512
- 64
- 256

What is printed when the following main method is executed?

```
public class Searcher
{
    private int[] arr = {1,3,5,8,9};

    public int mystery(int low, int high, int num) {
        int mid = (low + high) / 2;
        if (low > high) {
            return -1;    }
        else if (arr[mid] < num) {
            return mystery(mid + 1, high, num);    }
        else if (arr[mid] > num) {
            return mystery(low, mid - 1, num);    }
        else    return mid;
    }

    public static void main(String[] args)
    {
        Searcher s = new Searcher();
        System.out.println(s.mystery(0,4,3));
    }
}
```

-1

3

0

1

2

Which of the following correctly defines an interface?

- I. `public class Timer {
 public void start();
 public void stop();
 public int getTime();
}`
- II. `public interface Timer {
 private void start();
 private void stop();
 private int getTime();
}`
- III. `public interface Timer {
 public void start();
 I only public void stop();
 public int getTime();
}`

- II only
- I and III only
- III only
- I only
- II and III only

The following incomplete method is intended to sort the array `elem` in ascending order.

```
public static void sort(int[] elem)
{
    for (int j = 0; j < elem.length - 1; j++)
    {
        int minIndex = j;

        for (/* missing code */)
        {
            if (elem [k] < elem [minIndex])
            {
                minIndex = k;
            }
        }
        int temp = elem[j];
        elem[j] = elem[minIndex];
        elem[minIndex] = temp;
    }
}
```

Which of the following could be used to replace `/* missing code */` in the code above so that the method always sorts the array `a` in ascending order?

- `int k = j + 1; k < elem.length; k++`
- `int k = j - 1; k > 0; k--`
- `int k = j; k < elem.length; k++`
- `int k = j - 1; k >= 0; k--`
- `int k = j; k >= 0; k--`

29

What is the difference between an interface and an abstract class?

- Abstract classes can declare abstract methods, but interfaces can not.
- Interfaces are used in the standard Java classes in package java.util, but not abstract classes.
- Abstract classes can have methods with bodies (code in them), but interfaces can not.
- Abstract classes can be extended, but interfaces can not.
- Abstract classes can be instantiated, while interfaces can not.

****NOTE: THIS QUESTION WAS WRITTEN ABOUT JAVA VERSIONS PRIOR TO 8!!! Statement is NOT true for 8**

30

Consider the following code segment:

```
int [][] mat = new int [3][4];
for (int row = 0; row < mat.length; row++)
{
    for (int col = 0; col < mat[0].length; col++)
    {
        if (row > col)
            mat[row][col] = 1;
        else if (row == col)
            mat[row][col] = 2;
        else
            mat[row][col] = 3;
    }
}
```

What are the contents of mat after the code segment has been executed?

- {{1, 1, 2, 3},{1, 2, 3, 3}, {2, 3, 3, 3}}
- {{2, 1, 1},{3, 2, 1},{3, 3, 2},{3, 3, 3}}
- {{2, 1, 1, 1},{3, 2, 1, 1},{3, 3, 2, 1}}
- {{2, 3, 3, 3},{1, 2, 3, 3},{1, 1, 2, 3}}
- {{2, 3, 3},{1, 2, 3},{1, 1, 2},{1, 1, 1}}

Given the following method.

```
public int test(int v)
{
    for (int i = 0; i < a.length; i++)
    {
        if (a[i] == v)
            return i;
        else return -1;
    }
}
```

What would test return if a = {0,2,3,4} and v = 2?

- 1
- 0
- 2
- The code will not compile
- 1

Given the following declarations.

```
public class A {  
    public void test(B x, C y) {}  
}
```

```
public class B extends A {  
}
```

```
public class C extends B {  
}
```

Also consider the following code that appears in a different class.

```
A a = new A();  
B b = new B();  
C c = new C();
```

Which of the following is a correct call to `test`?

- `c.test(b,b);`
- `a.test(c,a);`
- `c.test(c,a);`
- `b.test(c,c);`
- `a.test(c,b);`

33

Which of the following would be the correct result from the following expression?

$$123_{10} - 12_8 + 101_2 + E_{16}$$

133

130

131

132

136

Consider the following class declaration.

```
public class Test
{
    private int value;

    public Test(int n)
    {
        value = n;
    }

    public void add(int amount)
    {
        value = value + amount;
    }

    public int getValue()
    {
        return value;
    }
}
```

The following code appears in another class. What is output?

```
Test a = new Test(200);
Test b = new Test(200);
Test c = a;

a.add(200);
System.out.println(a.getValue() + " " + b.getValue() +
                    " " + c.getValue());
```

- 400 400 400
- 400 200 400
- 400 200 200
- 200 200 200
- 400 400 200

Consider the following partial class declaration.

```
public class Person implements Comparable
{
    private String first; // first name
    private String last;  // last name

    public int compareTo(Object test) {
        {
            // implementation not shown
        }

        // . . .
    }
}
```

Assume that the `Person` objects are ordered by last name and then first name. Which of the following will correctly implement `compareTo` for the `Person` class?

I.

```
public int compareTo(Object test) {
    Person testP = (Person) test;
    return (last.compareTo(testP.last) +
           first.compareTo(testP.first));
}
```

II.

```
public int compareTo(Object test)
{
    Person testP = (Person) test;
    if (first.compareTo(testP.first) == 0)
        return last.compareTo(testP.last);
    else
        return first.compareTo(testP.first);
}
```

III.

```
public int compareTo(Object test)
{
    Person testP = (Person) test;
    if (last.compareTo(testP.last) == 0)
        return first.compareTo(testP.first);
    else
        return last.compareTo(testP.last);
}
```

I and III only

I and II only

I only

III only

II only

Consider the following code that is part of a class declaration.

```
private int [] myStuff;

//precondition: myStuff contains
// integers in no particular order
public int mystery(int num)
{
    for (int k = myStuff.length - 1;
         k >= 0; k--)
    {
        if (myStuff[k] > num)
            return k;
    }
    return -1;
}
```

Which of the following best describes the contents of `myStuff` after the following statement has been executed?

```
int m = mystery(n);
```

- All values in position 0 through `m` are less than `n`.
- All values in position `m+1` through `myStuff.length-1` are $\leq n$.
- All values in positions `m+1` through `myStuff.length-1` are $\geq n$.
- The largest value that is smaller than `n` is at position `m`.
- The smallest value is at position `m`.

Consider the following declaration of the class `RandomList`, which has a constructor that is intended to initialize the instance variable `numList` to an `ArrayList` of `n` integer values in the range [0 to 10].

```
public class RandomList
{
    private List<Integer> numList;

    // precondition: n > 0
    // postcondition: numList has been initialized to an
    // ArrayList of length n; each element of numList
    // contains an Integer in the range of [0 to 10]
    public RandomList(int n)
    {
        /* missing code */
    }
}
```

Which of the following could be used to replace `/* missing code */` so that the constructor will work as intended?

```
I. numList = new ArrayList<Integer>();
   for (int k = 0; k < n; k++)
       numList.add((int) (Math.random() * 10));
```

```
II. numList = new ArrayList<Integer>();
   for (int k = 0; k < n; k++)
       numList.add((int) (Math.random() * 10) + 1);
```

```
III. numList = new ArrayList<Integer>();
   for (int k = 0; k < n; k++)
       numList.add((int) (Math.random() * 11));
```

- I only
- II only
- III only
- II and III
- None of the above

Consider the following code segment

```
public static void test(int[] a, int y)
{
    if (a.length > 1)
        a[1] = a[1] * 2;
    y = y * 2;
}
```

What are the values of `s` and `b` after the following has executed?

```
int[] s = {3,4};
int b = 4;
test(s,b);
```

- `s={6, 8}; b=8;`
- `s={3, 8}; b=8;`
- `s={6, 8}; b=4;`
- `s={3, 8}; b=4;`
- `s={3, 4}; b=4;`

The following method should sort the contents of the array elements. Which of the following should you replace `/* missing code */` so that it works correctly?

```
public static void sort(int[] elem)
{
    for (int j = 1; j < elem.length; j++)
    {
        int temp = elem [j];
        int pIndex = j;
        while (/* missing code */)
        {
            elem [pIndex] = elem[pIndex - 1];
            pIndex--;
        }
        elem[pIndex] = temp;
    }
}
```

- elem[pIndex - 1] && pIndex > 0
- elem[pIndex - 1] || pIndex > 0
- pIndex > 0 && temp > elem[pIndex - 1]
- pIndex > 0 || temp < elem[pIndex - 1]
- pIndex > 0 && temp < elem[pIndex - 1]

What is the output from `mystery(4321)` when `mystery` is defined as follows:

```
//precondition: x >=0
public static void mystery (int x) {

    if ((x / 10) != 0) {
        mystery(x / 10);
    }
    System.out.print(x % 10);
}
```

- 4321
- 12344321
- 32144123
- 43211234
- 1234