

1. What values are stored in arr after the following code segment has been executed?

```
int[] arr = {1, 2, 3, 4, 5, 6, 7, 8};
for (int k = 1; k<=6; k+=2)
{
    arr[7] = arr[k];
    arr[k] = arr[k+1];
    arr[k+1] = arr[7];
}
```

- A. 1 3 4 5 6 7 2 2
- B. 1 3 2 5 4 7 6 8
- C. 1 3 2 5 4 7 6 6
- D. 2 1 4 3 6 5 7 8
- E. 2 1 4 3 6 5 7 5

2. Consider the following class.

```
public class Game {
    private static int bestScore;
    private int score;
    private String player;
    < constructors and methods not shown >
}
```

Which of the following constructors or methods in Game will NOT cause a syntax error?

- I. `public void resetScore()`
`{ score = 0; bestScore = 100; }`
 - II. `public Game()`
`{ score = 0; bestScore = 0; }`
 - III. `public static void setPlayer(String name)`
`{ player = name; }`
- a) I only
 - b) II only
 - c) I and II only
 - d) II and III only
 - e) I, II, and III

3. Consider the following program:

```
public class Tester
{
    public void someMethod(int a, int b) {
        int temp = a;
        a = b;
        b = temp;
    }
}

public class TesterMain
{
    public static void main(String[] args)
    {
        int x = 6, y = 8;
        Tester tester1 = new Tester();
        tester1.someMethod(x,y);
    }
}
```

What is contained in x, y, and temp respectively right after last line in main?

- a) 6, 8, 6
- b) 8, 6, 6
- c) 8, 6, ? (where ? means undefined)
- d) 6, 8, ? (where ? means undefined)
- e) 8, 6, 8

4. What is displayed by this code segment?

```
int [] arr = new int[4];
for (int k=0;k < arr.length; k++)
    arr[k] = 3*k;

for (int k=0;k<arr.length;k++)
    System.out.print(arr[k]+ " " + arr.length+ " ");
```

- a) 0 1 3 1 6 1 9 1
- b) 0 4 3 4 6 4 9 4
- c) 0 0 3 3 6 6 9 9
- d) 0 3 3 3 6 3 9 3
- e) none of these

5. Look at the following code and choose the best answer:

```
public class Runner {
    public static void main( String [] args ) {
        int [] c = null;
        met( c );
    }
    private static void met( int[] a ) {
        System.out.println( a.length );
    }
}
```

- a) This code will not compile. The error will be: local variable c may not have been initialized
- b) This code will compile but if you try to run it, the following run-time exception will occur: NullPointerException
- c) This code will compile but if you try to run it, the following run-time exception will occur: variable c has not been initialized
- d) This code will compile and run. It will display a zero.

6. Consider the following program:

```
public class Tester {
    private int a = 0;
    private int b = 0;
    public void method1(int x, int y) {
        int temp = x;
        a = y+3;
        b = temp + 3;
    }
    public int getA() { return a;}
    public int getB() { return b;}
}

public class TesterMain
{
    public static void main(String[] args)
    {
        int x = 6, y = 8;
        Tester t1 = new Tester();
        t1.method1(x,y);
        System.out.print(t1.getA()+" "+t1.getB());
    }
}
```

- a) 8 6
- b) 0 0
- c) 6 8
- d) 9 11
- e) none of these

7. Consider the following code:

```
public class SomeClass {
    private static int x=5;
    private int y;

    public SomeClass(int n) {
        x+=2;
        y=n;
    }

    public int getX(){
        return x+1;
    }

    public int getY() {
        return y;
    }
}

public class SomeRunner {
    public static void main(String[] args) {
        SomeClass d1 = new SomeClass(5);
        SomeClass d2 = new SomeClass(-5);
        SomeClass d3 = new SomeClass(2);
        System.out.print(d1.getX()+" ");
        System.out.print(d2.getX()+" ");
        System.out.print(d1.getY()+" ");
        System.out.print(d2.getY()+" ");
    }
}
```

What is printed as a result of running SomeRunner?

- A) The SomeClass class will not compile because getX method must be static.
- B) 7 -3 5 -5
- C) 12 12 5 -5
- D) 11 11 5 -5

8) Select the best statement based on the code below:

```
Monkey[] cageOMonkeys = new Monkey[5];
```

- I An array able to hold 5 Monkey objects is instantiated.
- II There are 5 Monkey objects in cageOMonkeys.
- III Monkey.length is 5

- a) I only
- b) I and II only
- c) all three are true
- d) I and III only
- e) none of these

9. Consider the following code. *sting* is an array of int values already defined and filled.

```
int n = sting.length - 1;
int sum = sting[ n ];
while ( n > 0 ) {
    n--;
    sum += sting[ n ];
}
```

- a) sum of all elements of the array sting are stored in sum
- b) sum of all elements of the array sting except sting[0] are stored in sum
- c) index out of bounds error occurs
- d) an infinite loop occurs
- e) sum of all elements of the array sting except first and last are stored in sum

10. The code below is meant to locate the index of the first negative integer in array *manta* which is defined and filled with integers.

```
int n = 0;
while (manta[ n ] >= 0 )
    n = n + 1;
int location = n;
```

variable location will hold the index of the first negative number.....

- a) never b) always c) if and only if *manta* contains at least one negative integer
- d) if and only if *manta* contains at least one non-negative integer e) none of these

11. Consider the BingoCard class :

```
public class BingoCard {
    private int[] card;

    /* constructs a BingoCard with 20 random digits 1 - 90. */
    public BingoCard()
        { /* implementation not shown */ }
}
```

Client code that simulates a bingo game declares an array of type BingoCard. The array has NUMPLAYERS elements, where each is a card of a different player. Here is a code segment that creates all bingo cards in the game.

```
/* declare array of BingoCard */
/* construct each BingoCard */
```

Which of the following is a correct replacement for `/* construct each BingoCard */`

- a) `for (int n = 1; n < players.length; n++)`
 `players[n] = new BingoCard();`
- b) `for (int n = 0; n < players.length(); n++)`
 `players[n] = new BingoCard();`
- c) `for (int n = 1; n < players.length(); n++)`
 `players[n] = new BingoCard();`
- d) `for (int n = 0; n < players.length; n++)`
 `players[n] = new BingoCard();`

12. Here are the private instance variables of a Frog object:

```
public class Frog {  
  
    private String species;  
    private int age;  
    private double weight;  
    private Position position;  
    private boolean amAlive;  
    .....
```

Which of the following methods in the Frog class is the best candidate for being a static method?

- a) getPondTemp // returns temperature of the pond
- b) makeSwim // frog swims to a new position in pond
- c) feedFrog // frog eats and gains weight
- d) getWeight // returns weight of frog
- e) killFrog // make this frog appear dead.

1. C 2. C 3. D 4. B 5. B 6. E 7. C 8. A 9. A 10. C 11. D 12. A