

**MULTIPLE CHOICE. Choose the one alternative that best completes the statement or answers the question.**

1) What is the output of the following code:

```
public class Test {  
    public static void main(String[] args) {  
        String s1 = new String("Java");  
        String s2 = new String("Java");  
        System.out.print((s1 == s2) + " " + (s1.equals(s2)));  
    }  
}
```

- A) false false
- B) true true
- C) false true
- D) true false
- E) None of the above

2) Given the following program:

```
public class Test {  
    public static void main(String[] args) {  
        for (int i = 0; i < args.length; i++) {  
            System.out.print(args[i] + " ");  
        }  
    }  
}
```

What is the output, if you run the program using

java Test 1 2 3

- A) 3
- B) 1 2 3
- C) 1 2
- D) 1
- E) None of the above

3) Which of the following statements represents the number of columns in a regular two-dimensional array named values?

- A) values[0].length
- B) values.length
- C) values.length()
- D) values[0].length()
- E) values.getColumnLength()

4) Analyze the following code:

```
public class Test {  
    public static void main(String[] args) {  
        A a = new A("3");  
        System.out.println(a.s);  
    }  
}  
  
public class A {  
    private String s;  
  
    public A(String s) {  
        this.s = s;  
    }  
  
    public void print() {  
        System.out.println(s);  
    }  
}
```

- A) The program has a compile error because System.out.println method cannot be invoked from a method.
- B) The program will display 3.
- C) The program has a compile error because s has not been initialized.
- D) The program has a compile error because you cannot create an object from the class that defines the object.
- E) The program has a compile error because s cannot be accessed directly outside class A.

5) What is the output of the following code?

```
int[] myList = {1, 5, 5, 5, 5, 1};  
int max = myList[0];  
int indexOfMax = 0;  
for (int i = 1; i < myList.length; i++) {  
    if (myList[i] > max) {  
        max = myList[i];  
        indexOfMax = i;  
    }  
}  
System.out.println(indexOfMax);
```

- A) 0
- B) 1
- C) 2
- D) 3
- E) 4

6) Given the classes below, show the output of the following program execution:

<pre>public class SmallNumberException extends Exception {     // .....     //Sets up exception object with a particular message.     // .....     SmallNumberException (String message)     {         super (message);     } }</pre>	<pre>public class test {     static public void main (String[] args)     {         final int MIN=0;         String[] values={"7", "3", "this", "5"};         SmallNumberException problem=new SmallNumberException("Too small.");          for (int i=0;i&lt;values.length;i++)         {             try{                 if (Integer.parseInt(values[i])&lt;MIN) throw problem;                 System.out.print(Integer.parseInt(values[i])%2+ " ");             }             catch (SmallNumberException e){                 System.out.println(e.getMessage());             }             catch (NumberFormatException e){                 System.out.println("not a number");             }         }     } }</pre>
---	--

**Answer:**

7) Analyze the following code:

```
public class Test {
    public static void main(String[] args) {
        int[] x = new int[5];
        int i;
        for (i = 0; i < x.length; i++)
            x[i] = i;
        System.out.println(x[i]);
    }
}
```

- A) The program displays 0 1 2 3 4.
- B) The program displays 4.
- C) The program has a runtime error because the last statement in the main method causes `ArrayIndexOutOfBoundsException`.
- D) The program has a compile error because `i` is not defined in the last statement in the main method.
- E) None of the above

8) What is the output of the following code?

```
class Test {  
    public static void main(String[] args) {  
        try {  
            System.out.println("Welcome to Java");  
            int i = 0;  
            int y = 2/i;  
            System.out.println("Welcome to Java");  
        }  
        catch (ArithmeticException e){  
            System.out.println("Exception handled");  
        }  
  
        System.out.println("End of the block");  
    }  
}
```

- A) The program displays Welcome to Java three times followed by End of the block.
- B) The program displays Welcome to Java two times followed by Exception handled.
- C) The program displays Welcome to Java two times followed by End of the block.
- D) The program displays Welcome to Java and End of the block.
- E) None of the above

9) Show the output of the following code:

```
public class Test {  
    public static void main(String[] args) {  
        int[] x = {1, 2, 3, 4, 5};  
        increase(x);  
  
        int[] y = {1, 2, 3, 4, 5};  
        increase(y[0]);  
  
        System.out.println(x[0] + " " + y[0]);  
    }  
  
    public static void increase(int[] x) {  
        for (int i = 0; i < x.length; i++)  
            x[i]++;  
    }  
  
    public static void increase(int y) {  
        y++;  
    }  
}
```

- A) 0 0
- B) 1 1
- C) 2 2
- D) 2 1
- E) 1 2

10) Consider the following class definitions:

```
public class Employee
{
    /*      ..      */
}

public class SalariedEmployee extends Employee
{
    /*      ..      */
}
```

Which of the following is incorrect?

- A) SalariedEmployee is a subclass of Employee
- B) Employee is a superclass of SalariedEmployee
- C) SalariedEmployee inherits all properties and methods of Employee
- D) Employee is a subclass of java.lang.Object
- E) Employee does not inherit from Object (example: equals())

**Write your answer in the space provided.**

- 11) Write the definition for a child class of **Circle** named **Cylinder**. The cylinder should have a double attribute named **height**. The constructor of the child class should take parameters to initialize the attributes. The class **Cylinder** should also include getter and setter method as well as an equals method that returns true if two cylinder objects have same dimensions and false otherwise. Remember to override method where needed.

```
public class Circle
{
    protected double radius;

    public Circle(double r) {
        radius=r;
    }

    public void setRadius(double r) {
        radius = r;
    }
    public double getRadius() {
        return radius;
    }
    public double area() {
        return Math.PI * radius * radius;
    }
    public String toString() {
        return "Circle:Radius= " + radius;
    }
}
```

12) Implement a method called **eStatistics** that takes an array of strings called **names**. The method returns an array that contains the frequencies of letter 'e' in every string in the original array.

For example, if

`names={"Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"}`

the method returns the array `{0, 1, 2, 0, 0, 0, 0}`

13) Implement a **recursive** method named **power** that takes 2 **integer** parameters named **base** and **expo**. The method will **return** the base raised to the power of expo.