1. What happens when you try to compile and run the following program?

public class MyStatic {

   static int x = 6;

   MyStatic() {

      x ++ ;

   }

   void method(){

      System.out.print("-x" + x);

   }

   public static void main(String[] args){

      MyStatic mc1,mc2,mc3,mc4;

      MyStatic mc5 = new MyStatic();

      MyStatic mc6 = new MyStatic();

      MyStatic mc7 = new MyStatic();

      mc7.method();

   }

}

Select the correct answer.

a. This program is illegal at runtime time.

b. This program writes “-x6” to the standard output.

c. This program writes “-x3” to the standard output.

d. This program writes “-x9” to the standard output.

e. This program writes “-x7” to the standard output.

f. This program writes “-x0” to the standard output.

g. This program writes “-x1” to the standard output.

h. This program writes nothing to the standard output.

1. What happens when the following program is compiled and run?

public class Counter {

   int count;

   static int stCount;

   public Counter() {

      count ++ ;

      stCount ++ ;

   }

   public int getCount(){

      return count;

   }

   public static int getStCount(){

      return stCount;

   }

}

public class TestCounter {

   public static void main(String[] args){

      Counter cs1 = new Counter();

      Counter cs2 = new Counter();

      Counter cs3 = new Counter();

      Counter cs4 = new Counter();

      Counter cs5 = new Counter();

      Counter cs6 = new Counter();

      System.out.println("count is: " + cs6.getCount());

      System.out.println("stCount is: " + cs6.getStCount());

   }

}

1. We used the sixth object to return the total numbers of the objects that have been instantiated. What would happen if you use the first object instead of the sixth one?

System.out.println("count is: "+cs1.getCount());

System.out.println("stCount is: "+cs1.getStCount());

1. 1- What is written to the standard output as the result of executing the following statements?

2- Is it necessary to instantiate the class Calculus? clarify your answer.

3- How can you prevent anyone to instantiate the class Calculus?

4- If you prevent instantiating the class Calculus, how can you access the method getSum?

public class Calculus {

   public static double getSum(double d1, double d2){

      return d1 + d2;

   }

}

public class TestCalculus {

   public static void main(String[] args){

      Calculus cal = new Calculus();

      Calculus cal2 = new Calculus();

      Calculus cal3 = new Calculus();

      System.out.println(cal.getSum(59,78));

      System.out.println(cal2.getSum(59,78));

      System.out.println(cal3.getSum(59,78));

   }

}