**Example 5(static variables)**

**Example part1:**

Find the output of the following:.

**public class Test {**

**public int numOfObjects = 0;**

**public Test()**

**{**

**numOfObjects++; // numOfObjects = numOfObjects+1**

**}**

**public static void main(String[] args) {**

**Test t1 = new Test();**

 **System.out.println(t1.numOfObjects);**

**Test t2 = new Test();**

 **System.out.println(t1.numOfObjects);**

**Test t3 = new Test();**

**System.out.println(t1.numOfObjects);**

**}**

**}**

**Output:**

**1**

**1**

**1**

**Example part2:**

Find the output of the code in part 1 after define **numOfObjects as static.**

**Output:**

**1**

**2**

**3**

Notes:

* When an object is created, it has its own copy of all instance variables.
* Unlike instance variables; static variables are shared among objects, so there exists only one copy of static variables.
* Static variables can be called using class name.

Ex: System.out.println(**Test1**.numOfObjects);

* Static variables are created when the program starts and destroyed when the program stops so we don't need to create object from its class to use it.
* Static variables can be accessed by calling with the class name ClassName.VariableName. Ex: System.out.println(**Test1**.numOfObjects);

public class Test2 {

 public static int numOfObjects =0;

 public Test2()

 {

 numOfObjects++;

 }

public class Test1 {

 public static void main(String[] args) {

 Test2 t1 = new Test2();

 System.out.println(Test2.numOfObjects);

 Test2 t2 = new Test2();

 System.out.println(Test2.numOfObjects);

 Test2 t3 = new Test2();

 System.out.println(Test2.numOfObjects);

 }

}

}



public class Test2 {

 public static int numOfObjects =0;

 public Test2()

 {

 numOfObjects++;

 }

}

public class Test1 {

 public static void main(String[] args) {

 System.out.println(Test2.numOfObjects);

 System.out.println(Test2.numOfObjects);

 System.out.println(Test2.numOfObjects);

 }

}

