Programming problem assignment

1.Create a class that keeps track of the number of instances created. Implement a static variable and method to accomplish this.

Answer:

class InstanceCount {

    private static int count = 0;

    public InstanceCount() {

        count++;

    }

    public static int getInstanceCount() {

        return count;

    }

}

public class Q5 {

    public static void main(String[] args) {

        InstanceCount obj1 = new InstanceCount();

        System.out.println("Number of instance : " + InstanceCount.getInstanceCount());

        InstanceCount obj2 = new InstanceCount();

        System.out.println("Number of instance : " + InstanceCount.getInstanceCount());

    }

}

// Create a class that keeps track of the number of instances created. Implement

// a static variable and method to accomplish this.

2.Write a program and create a constructor with parameters and initialise the variable using a constructor.

class Test {

    Test(double d) {

        System.out.println("double argument constructor");

    }

    Test(int i) {

        // this(10.5);

        System.out.println("int argument constructor");

    }

    Test() {

        this(10);

        System.out.println("no argument constructor");

    }

}

public class Q2 {

    public static void main(String[] args) {

        Test t1 = new Test();

        Test t2 = new Test(10);

        Test t3 = new Test(10.5);

    }

}

3.Use a private keyword for a variable and use setter and getter methods to initialise and print the values.

public class Q4 {

    private int num1;

    private int num2;

    // setter method

    void setValue() {

        System.out.println("We are in setter method");

        num1 = 10;

        num2 = 20;

    }

    // getter method

    int getSum() {

        System.out.println("we are in getter method");

        int sum = num1 + num2;

        return sum;

    }

    public static void main(String[] args) {

        Q4 obj = new Q4();

        obj.setValue();

        int r = obj.getSum();

        System.out.println("Sum is :" + r);

    }

}

4.Write a program to call an method without creating an object of a class

public class Q6 {

    static int num1;

    static int num2;

    static {

        num1 = 10;

        num2 = 20;

    }

    static void dispSum() {

        int sum = num1 + num2;

        System.out.println("Sum is : " + sum);

    }

    public static void main(String[] args) {

        dispSum();

    }

}

//Write a program to call an method without creating an object of a class

5.Write a program which has static block and constructor overloading.initialise variables using constructors and print it.

public class Program {

    private static String staticVariable;

    static {

        staticVariable = "Initialized in static block";

    }

    private String instanceVariable;

    public Program() {

        instanceVariable = "Initialized in default constructor";

    }

    public Program(String value) {

        instanceVariable = "Initialized in parameterized constructor with value: " + value;

    }

    public static void main(String[] args) {

        Program obj1 = new Program();

        System.out.println("Instance variable value (obj1): " + obj1.instanceVariable);

        System.out.println("Static variable value: " + staticVariable);

        Program obj2 = new Program("Hello");

        System.out.println("Instance variable value (obj2): " + obj2.instanceVariable);

        System.out.println("Static variable value: " + staticVariable);

    }

}

// Write a program which has static block and constructor overloading.initialise

// variables using constructors and print it.