OOP Lab

Week 2 Assignment Submission

Lab 2 : Classes and Objects, Constructors and Static Members

1) Define a class to represent a complex number called Complex. Provide the following methods:

1. To assign initial values to the Complex object.
2. To display a complex number in a+ib format.
3. To add 2 complex numbers. (the return type should be Complex)
4. To subtract 2 complex numbers Write a main method to test the class.

Code:

```
import java.util.*;

public class Complex

{

   int a,ib;

   Complex()   //Default Constructor

   {

      this.a=0;

      this.ib=0;

   }

   Complex(int c,int d) //Parameterized Constructor
```

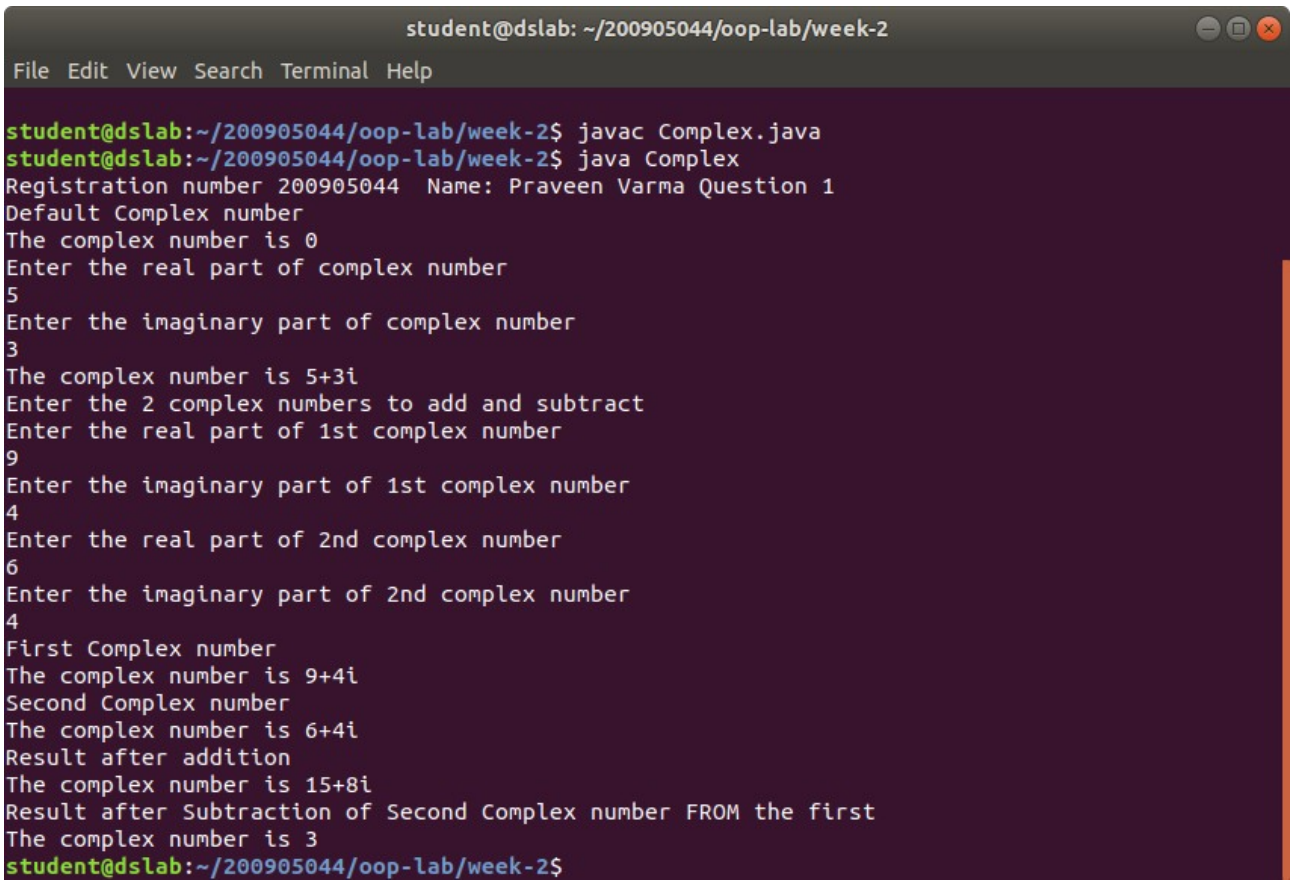
```
    {
       this.a=c;
       this.ib=d;
    }
    public void Display()
    {
       if(ib==0)
       System.out.println("The complex number is "+a);
       else if(ib!=0 && a==0)
       System.out.println("The complex number is "+ib+"i");
       else
       System.out.println("The complex number is "+a+"+"+ib+"i");
    }
    public Complex Add(Complex Cmp1,Complex Cmp2)
    {
       int x=Cmp1.a+Cmp2.a;
       int y=Cmp1.ib+Cmp2.ib;
       Complex ResAdd=new Complex(x,y);    //Generating new complex
number after addition
       return ResAdd;
    }
```

```
    public Complex Subtract(Complex Cmp1,Complex Cmp2)

    {

        int x=Cmp1.a-Cmp2.a;

        int y=Cmp1.ib-Cmp2.ib;

        Complex ResSub=new Complex(x,y);  //Generating new complex
number after subtraction

        return ResSub;

    }

    public static void main(String[] args)

    {

        System.out.println("Registration number 200905044  Name:
Praveen Varma Question 1");

        Complex cmp=new Complex(); // Generating new Complex number
using default constructor

        System.out.println("Default Complex number");

        cmp.Display(); //Displaying Default value of Complex number

        Scanner sc=new Scanner(System.in);



        System.out.println("Enter the real part of complex number");

        int Real=sc.nextInt();

        System.out.println("Enter the imaginary part of complex number");

        int Imag=sc.nextInt();
```

```
        cmp=new Complex(Real,Imag); // Generating new Complex number
using parameterised constructor

        cmp.Display(); //Displaying the complex number entered by the user

        System.out.println("Enter the 2 complex numbers to add and
subtract");

        System.out.println("Enter the real part of 1st complex number");



        int a=sc.nextInt();

        System.out.println("Enter the imaginary part of 1st complex
number");

        int b=sc.nextInt();

        Complex Cmp1=new Complex(a,b); // Generating first Complex
number using parameterised constructor



        System.out.println("Enter the real part of 2nd complex number");

        int x=sc.nextInt();

        System.out.println("Enter the imaginary part of 2nd complex
number");

        int y=sc.nextInt();



        Complex Cmp2=new Complex(x,y); // Generating second Complex
number using parameterised constructor

        System.out.println("First Complex number");
```

```
        Cmp1.Display();

        System.out.println("Second Complex number");

        Cmp2.Display();

        Complex ResAdd=new Complex(); // Generating Resultant Complex
number using parameterised constructor

        ResAdd=ResAdd.Add(Cmp1,Cmp2);

        System.out.println("Result after addition");

        ResAdd.Display(); // displaying result of addition

        Complex ResSub=new Complex(); // Generating Resultant Complex
number using parameterised constructor

        ResSub=ResSub.Subtract(Cmp1,Cmp2);

        System.out.println("Result after Subtraction of Second Complex
number FROM the first");

        ResSub.Display(); // displaying result of subtraction



    }

}
```

Output:



2. Create a class called Time that has instance variables to represent hours, minutes and seconds. Provide the following methods:
    1. To assign initial values to the Time object.
    2. To display a Time object in the form of hh:mm:ss {24 hours format}
    3. To add 2 Time objects (the return type should be a Time)
    4. To subtract 2 Time objects (the return type should be a Time)
    5. To compare 2 Time objects and to determine if they are equal or if the first is greater or smaller than the second one.

Code:

```
import java.util.*;
```

```
import java.lang.*;



public class Time {

    int Hours, Mins, Secs;


    // default constructor

    Time() {

        Hours = 0;

        Mins = 0;

        Secs = 0;

    }

 // parameterised constructor

    Time(int h, int m, int s) {

        this.Hours = h;

        this.Mins = m;

        this.Secs = s;

    }


    // display the Time object in hh:mm:ss format
    void Display() {
```

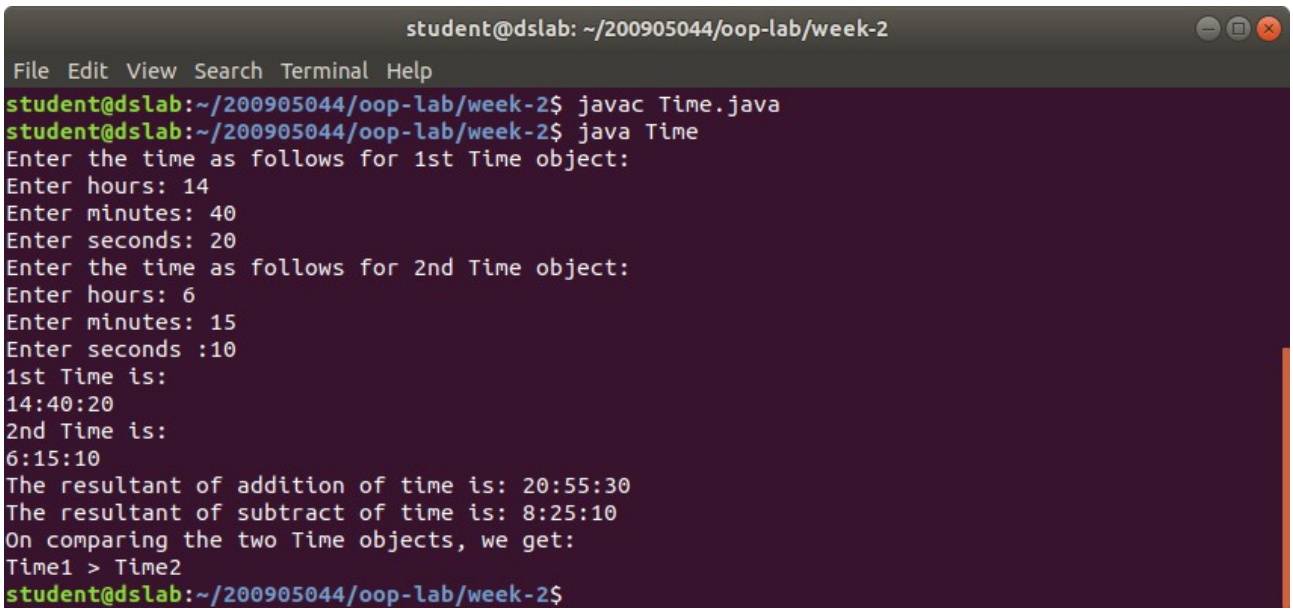
```
    System.out.println(+this.Hours + ":" + this.Mins + ":" + this.Secs);

}



// add two Time objects
Time Add(Time t1, Time t2) {

    Time t3 = new Time();

    t3.Hours = t1.Hours + t2.Hours;

    t3.Mins = t1.Mins + t2.Mins;

    t3.Secs = t1.Secs + t2.Secs;

    if (t3.Secs >= 60) {

        t3.Secs %= 60;

        t3.Mins += 1;

    }

    if (t3.Mins >= 60) {

        t3.Mins %= 60;

        t3.Hours += 1;

    }

    if (t3.Hours > 23) {

        t3.Hours %= 24;

    }

    return t3;
```

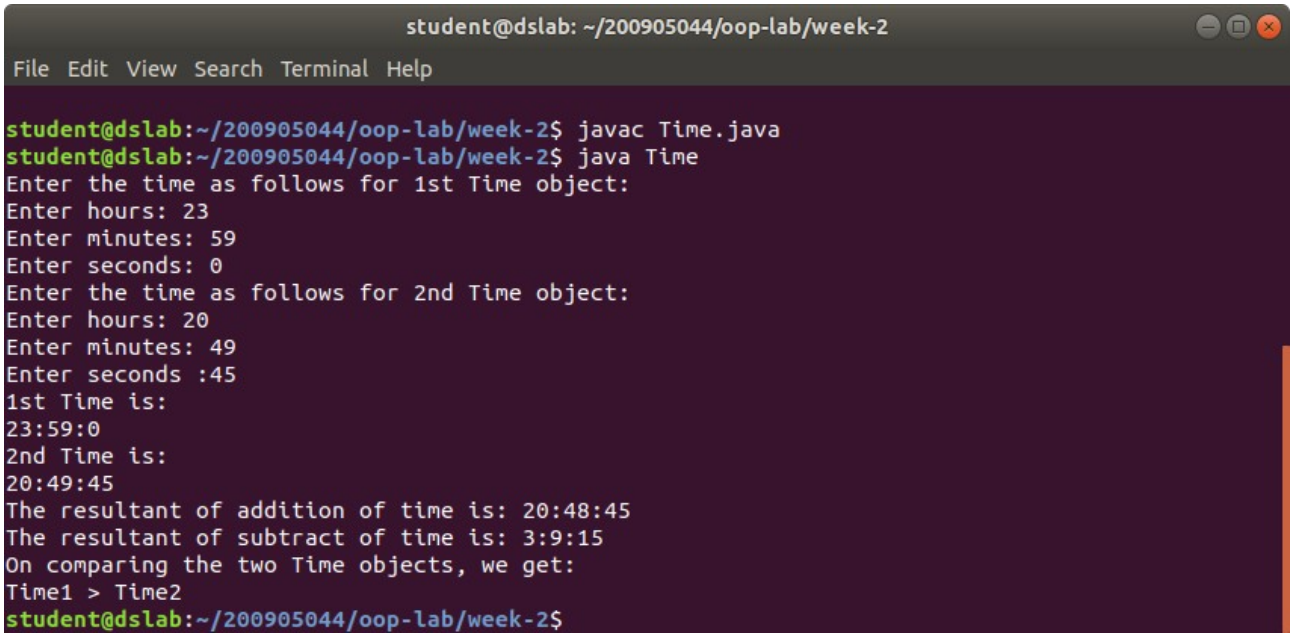
```
    }


    // subtract two Time objects
    Time Subtract(Time T1, Time T2) {

        Time T3 = new Time();

        int t1, t2, diff;

        t1 = T1.Secs + T1.Mins * 60 + T1.Hours * 60 * 60;

        t2 = T2.Secs + T2.Mins * 60 + T2.Hours * 60 * 60;

        diff = Math.abs(t1 - t2);

        T3.Hours = diff / 3600;

        diff = diff % 3600;

        T3.Mins = diff / 60;

        T3.Secs = diff % 60;

        return T3;

    }



    void Compare(Time t1, Time t2) {

        if (t1.Hours == t2.Hours && t1.Mins == t2.Mins && t1.Secs ==
t2.Secs)

            System.out.println("Time = Time2");
```

```
        else if (t1.Hours > t2.Hours || (t1.Hours == t2.Hours && t1.Mins >
t2.Mins)

                || (t1.Hours == t2.Hours && t1.Mins == t2.Mins && t1.Secs >
t2.Secs))

            System.out.println("Time1 > Time2");

        else

            System.out.println("Time1 < Time2");

    }


    public static void main(String[] args) {


        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the time as follows for 1st Time object:");

        System.out.print("Enter hours: ");
        int h = sc.nextInt();

        System.out.print("Enter minutes: ");
        int m = sc.nextInt();

        System.out.print("Enter seconds: ");
        int s = sc.nextInt();

        Time ob1 = new Time(h,m,s);

        System.out.println("Enter the time as follows for 2nd Time object:");

        System.out.print("Enter hours: ");
        h = sc.nextInt();

        System.out.print("Enter minutes: ");
        m = sc.nextInt();
```

```
        System.out.print("Enter seconds :");
        s = sc.nextInt();

        Time ob2 = new Time(h,m,s);

        System.out.println("1st Time is: ");
        ob1.Display();

        System.out.println("2nd Time is: ");
        ob2.Display();

        Time ob3 = new Time();


        ob3=ob3.Add(ob1,ob2);
        System.out.println("The resultant of addition of time is:
"+ob3.Hours+":"+ob3.Mins+":"+ob3.Secs);


        ob3=ob3.Subtract(ob1,ob2);
        System.out.println("The resultant of subtract of time is:
"+ob3.Hours+":"+ob3.Mins+":"+ob3.Secs);


        System.out.println("On comparing the two Time objects, we get: ");
        ob3.Compare(ob1,ob2);

    }

}
```

Output:





3. Consider the already defined Complex class. Provide a default constructor and parameterized constructor to this class. Also provide a display method. Illustrate all the constructors as well as the display method by defining Complex objects.

Code:

```
import java.util.*;
```

```
public class Complex
{
   int a,ib;
   Complex()   //Default Constructor
   {
      this.a=0;
      this.ib=0;
   }
   Complex(int c,int d) //Parameterized Constructor
   {
      this.a=c;
      this.ib=d;
   }
 // displaying the complex number
   public void Display()
   {
      if(ib==0)
      System.out.println("The complex number is "+a);
      else if(ib!=0 && a==0)
      System.out.println("The complex number is "+ib+"i");
```

```
        else
        System.out.println("The complex number is "+a+"+"+ib+"i");
    }
    public Complex Add(Complex Cmp1,Complex Cmp2)
    {
        int x=Cmp1.a+Cmp2.a;
        int y=Cmp1.ib+Cmp2.ib;
        Complex ResAdd=new Complex(x,y);    //Generating new
complex number after addition
        return ResAdd;
    }
    public Complex Subtract(Complex Cmp1,Complex Cmp2)
    {
        int x=Cmp1.a-Cmp2.a;
        int y=Cmp1.ib-Cmp2.ib;
        Complex ResSub=new Complex(x,y);  //Generating new
complex number after subtraction
        return ResSub;
    }
    public static void main(String[] args)
    {
```

System.out.println("Registration number 200905044 Name: Praveen Varma Question 3");

Complex cmp=new Complex(); **// Generating new Complex number using default constructor**

System.out.println("Default Complex number");

cmp.Display(); **//Displaying Default value of Complex number**

Scanner sc=new Scanner(System.in);

System.out.println("Enter the real part of complex number");

int Real=sc.nextInt();

System.out.println("Enter the imaginary part of complex number");

int Imag=sc.nextInt();

cmp=new Complex(Real,Imag); **// Generating new Complex number using parameterised constructor**

cmp.Display(); **//Displaying the complex number entered by the user**

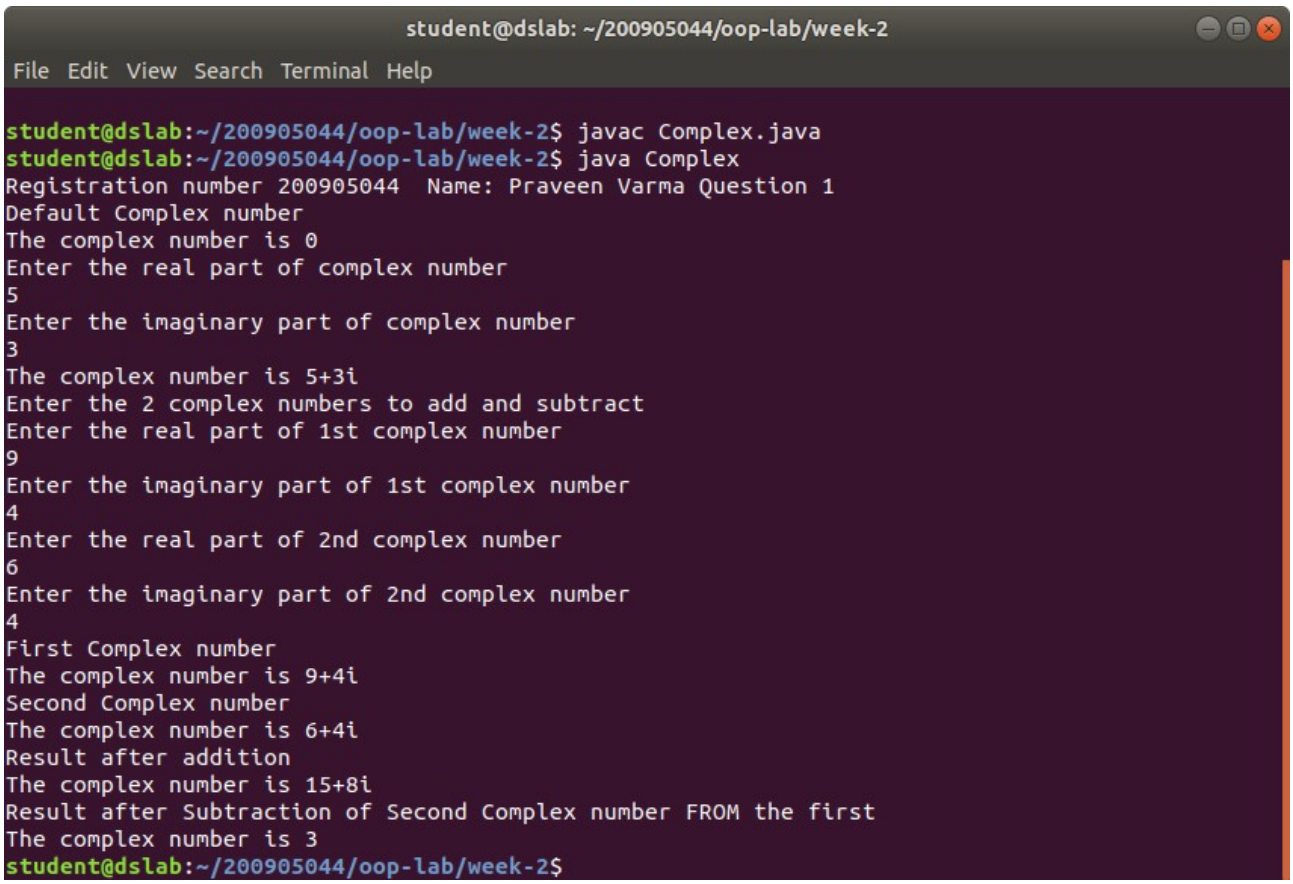
System.out.println("Enter the 2 complex numbers to add and subtract");

System.out.println("Enter the real part of 1st complex number");

```
        int a=sc.nextInt();

        System.out.println("Enter the imaginary part of 1st complex
number");

        int b=sc.nextInt();

        Complex Cmp1=new Complex(a,b); // Generating first
Complex number using parameterised constructor



        System.out.println("Enter the real part of 2nd complex
number");

        int x=sc.nextInt();

        System.out.println("Enter the imaginary part of 2nd complex
number");

        int y=sc.nextInt();



        Complex Cmp2=new Complex(x,y); // Generating second
Complex number using parameterised constructor

        System.out.println("First Complex number");

        Cmp1.Display();

        System.out.println("Second Complex number");

        Cmp2.Display();

        Complex ResAdd=new Complex(); // Generating Resultant
Complex number using parameterised constructor
```

```
        ResAdd=ResAdd.Add(Cmp1,Cmp2);

        System.out.println("Result after addition");

        ResAdd.Display(); // displaying result of addition

        Complex ResSub=new Complex(); // Generating Resultant
Complex number using parameterised constructor

        ResSub=ResSub.Subtract(Cmp1,Cmp2);

        System.out.println("Result after Subtraction of Second
Complex number FROM the first");

        ResSub.Display(); // displaying result of subtraction



    }

}
```

Output:



4. Create a class called Counter that contains a static data member to count the number of Counter objects being created. Also define a static member function called showCount() which displays the number of objects created at any given point of time. Illustrate this.

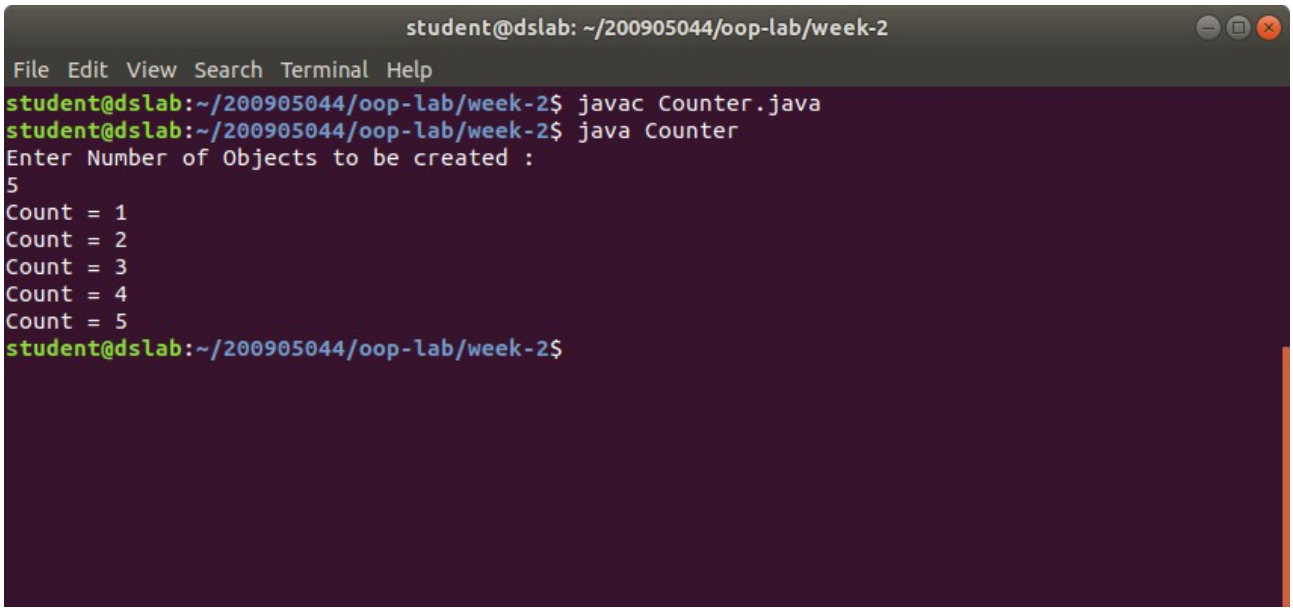
Code:

```
import java.util.Scanner;

class Counter

{
```

```
static int count=0;


Counter()
{
    count++;
}


static void showCount()
{
    System.out.println("Count = "+count);
}


public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter Number of Objects to be created : ");
    int n = sc.nextInt();

    Counter obj[] = new Counter[n];


    for(int i = 0 ; i < n ; i++){
        obj[i] = new Counter();
        obj[i].showCount();
    }
}
```

}

Output:



THANK YOU!