SESSION 1

PART 2 :

LAB 3 :

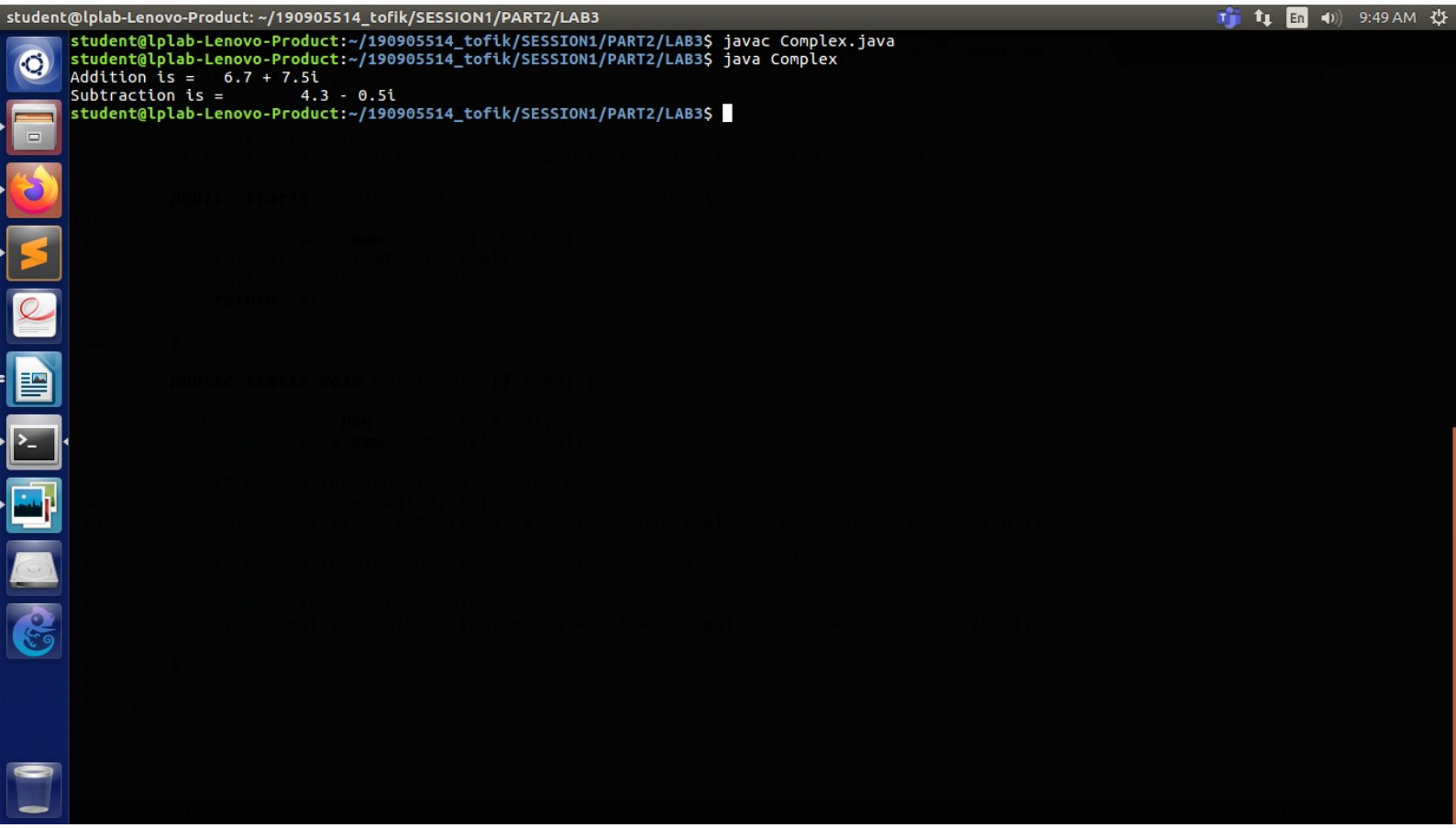
Define a class to represent a complex number called Complex. Provide the following methods:

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.

1->

```
public class Complex {

    double real;
    double imag;

    public Complex(double r, double img) {

        this.real = r;
        this.imag = img;
    }
    /* Add() for two complex number */


    public static Complex add(Complex n1, Complex n2) {

        Complex complex = new Complex(0.0, 0.0);
        complex.real = n1.real + n2.real;
        complex.imag = n1.imag + n2.imag;
        return complex;
    }

    /* Substracting tow number here */


    public static Complex sub(Complex n1, Complex n2) {

        Complex sb = new Complex(0.0, 0.0);
        sb.real = n1.real - n2.real;
        sb.imag = n1.imag - n2.imag;
        return sb;

    }

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

```
        Complex z1 = new Complex(5.5, 4);
         Complex z2 = new Complex(1.2, 3.5);

         /* Displaying add of two complex number */
         Complex complex = add(z1, z2);
         System.out.printf("Addition is = \t" + complex.real + " +
" + complex.imag + "i\n");

         /* Displaying the substracting of two complex number */

         Complex sb = sub(z1, z2);
         System.out.printf("Subtraction is = \t" + sb.real + " - "
+ sb.imag + "i\n");

    }

}
```



LAB 3 :

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}

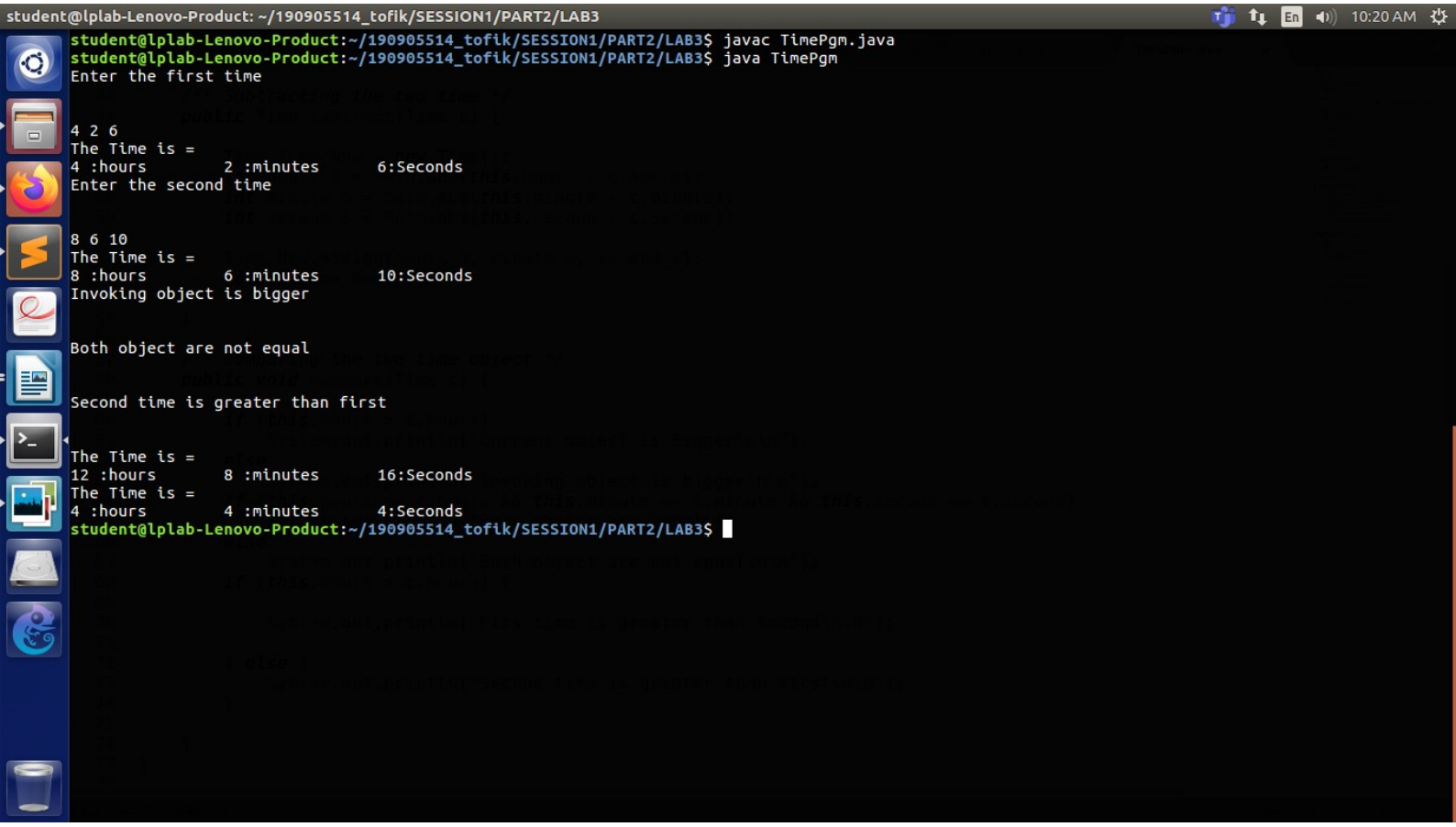
3. To add 2 Time objects (the return type should be a Time)
4. To subtract 2 Time objects (the return type should be a Time)
5. To compare 2 Time objects and to determine if they are equal or if the first is greater or smaller than the second one.

2->

```
import java.util.Scanner;

class Time {

    int hours;
    int minute;
    int second;

    public void assign(int a, int b, int c) {

        this.hours = a;
        this.minute = b;
        this.second = c;
    }

    public void Display() {

        System.out.println("The Time is = \n" + "" + this.hours +
" :hours\t" + this.minute + " :minutes\t" +
this.second+":Seconds");
    }

    /** Adding the time */
    public Time add(Time c) {
        Time Time_New = new Time();
        int hours_h = this.hours + c.hours;
        int minute_m = this.minute + c.minute;
        int second_s = this.second + c.second;

        if (hours_h > 24) {
            hours_h = 1;
        }
        if (minute_m > 60) {
            hours_h++;
        }
        if (second_s > 60) {
            minute_m++;
        }

        Time_New.assign(hours_h, minute_m, second_s);

        return Time_New;
```

```
    }

    /** Subtracting the two time */
    public Time subtract(Time c) {

        Time Time_New = new Time();
        int hours_h = Math.abs(this.hours - c.hours);
        int minute_m = Math.abs(this.minute - c.minute);
        int second_s = Math.abs(this.second - c.second);

        Time_New.assign(hours_h, minute_m, second_s);
        return Time_New;

    }

    /** comparing the two time object */
    public void compare(Time c) {

        if (this.hours > c.hours)
            System.out.println("Current object is Bigger\n\n");
        else
            System.out.println("Invoking object is bigger\n\n");
        if (this.hours == c.hours && this.minute == c.minute &&
this.second == c.second)
            System.out.println("Both object are equal");
        else
            System.out.println("Both object are not equal\n\n");
        if (this.hours > c.hours) {

            System.out.println("Firs time is greater than second\
n\n");

        } else {
            System.out.println("Second time is greater than first\
n\n");
        }

    }
}

public class TimePgm {
    public static void main(String[] args) {

        int x;
        int y;
        int z;
        int a;
        int b;
        int c;
        Scanner sc = new Scanner(System.in);
        Time c1 = new Time();
        Time c2 = new Time();
```

```
        System.out.println("Enter the first time \n\n");
        x = sc.nextInt();
        y = sc.nextInt();
        z = sc.nextInt();

        c1.assign(x, y, z);
        c1.Display();

        System.out.println("Enter the second time\n\n");

        a = sc.nextInt();
        b = sc.nextInt();
        c = sc.nextInt();
        c2.assign(a, b, c); /* Assigning the value of date */
        c2.Display(); /* Displaying the time */


        c1.compare(c2);


        Time add_new = new Time();
        add_new = c1.add(c2);
        Time student_time = new Time();
        student_time = c1.subtract(c2);
        add_new.Display();
        student_time.Display();

    }

}
```



LAB 4 :

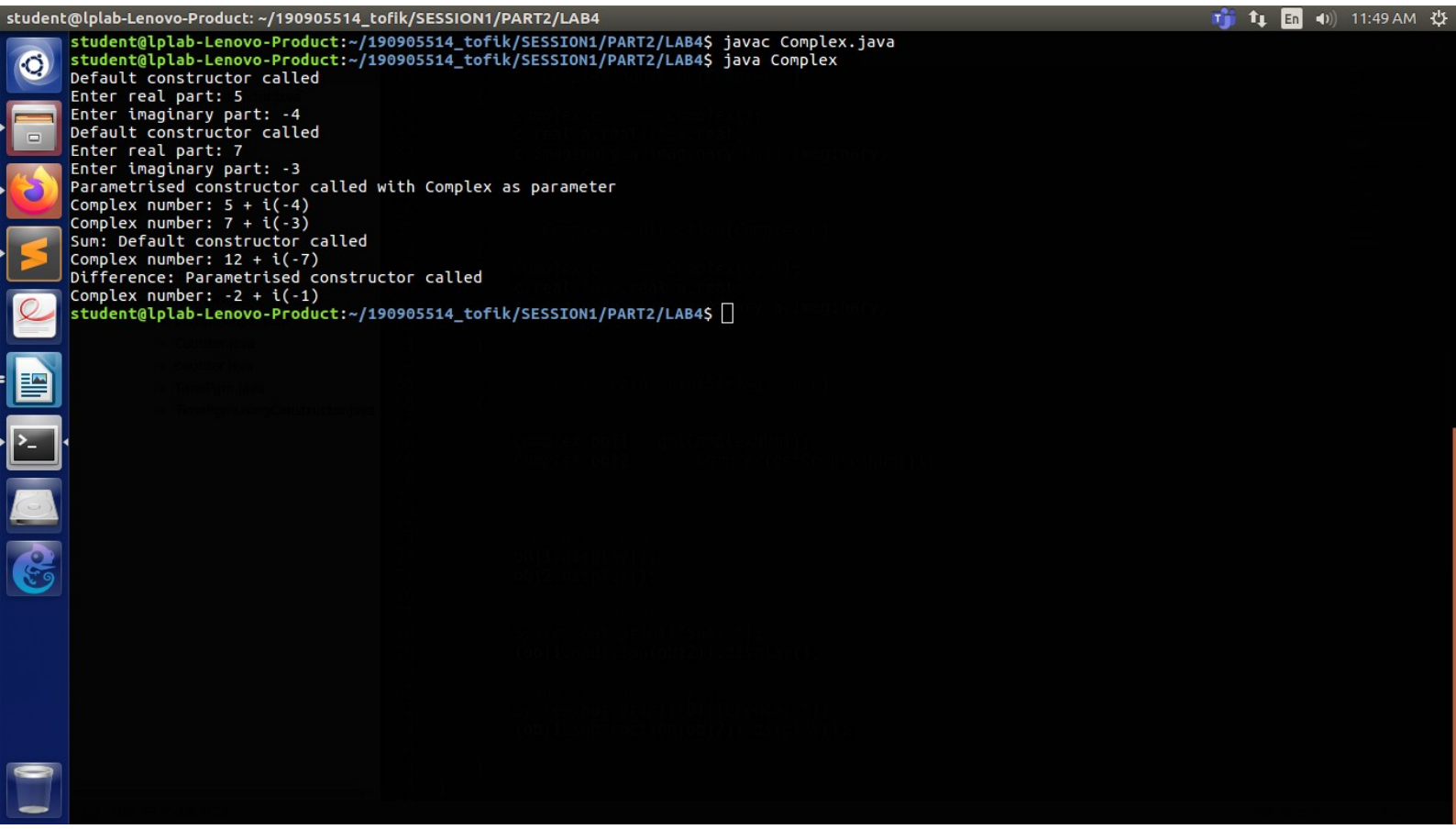
Consider the already defined Complex class. Provide a default constructor and parameterized constructor to this class. Also provide a display method. Illustrate all the constructors as well as the display method by defining Complex objects.

1->

```
import java.util.Scanner;
import java.lang.Math;
```

```
public class Complex
{
     int real, imaginary;


     public Complex()
     {
          System.out.println("Default constructor called");
          this.real=0;
          this.imaginary=0;
     }


     public Complex(int re, int im)
     {
          System.out.println("Parametrised constructor called");
          this.real=re;
          this.imaginary=im;
     }


     public Complex(Complex c)
     {
          System.out.println("Parametrised constructor called with
Complex as parameter");
          this.real=c.real;
          this.imaginary=c.imaginary;
     }

     public static Complex getComplexNum()
     {
          Complex a = new Complex();
          Scanner sc=new Scanner(System.in);
          System.out.print("Enter real part: ");
          a.real=sc.nextInt();
          System.out.print("Enter imaginary part: ");
          a.imaginary=sc.nextInt();

          return a;
     }

     public void display()
     {
          System.out.println("Complex number: "+this.real+" +
i("+this.imaginary+")");
     }

     public Complex addition(Complex a)
     {
          Complex c = new Complex();
          c.real=a.real+this.real;
```

```
        c.imaginary=a.imaginary+this.imaginary;
        return c;
    }

    public Complex subtraction(Complex a)
    {
        Complex c = new Complex(0, 0);
        c.real=this.real-a.real;
        c.imaginary=this.imaginary-a.imaginary;
        return c;
    }

    public static void main(String[] arg)
    {

        Complex obj1 = getComplexNum();
        Complex obj2 = new Complex(getComplexNum());



        //Display method
        obj1.display();
        obj2.display();

        //Add and display
        System.out.print("Sum: ");
        (obj1.addition(obj2)).display();

        //Subtract and display
        System.out.print("Difference: ");
        (obj1.subtraction(obj2)).display();

    }
}
```



LAB 4 :

2->

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.

```
import java.util.Scanner;
class Counter
{
    static int n;

    static void showCounter()
    {
        System.out.println(n);
    }

    Counter()
    {
        n = n + 1;
    }

}
class counter
```

```
{
    public static void main(String args[])
    {
        Counter c1 = new Counter();
        Counter c2 = new Counter();
        Counter c3 = new Counter();
        c3.showCounter();
    }
}
```