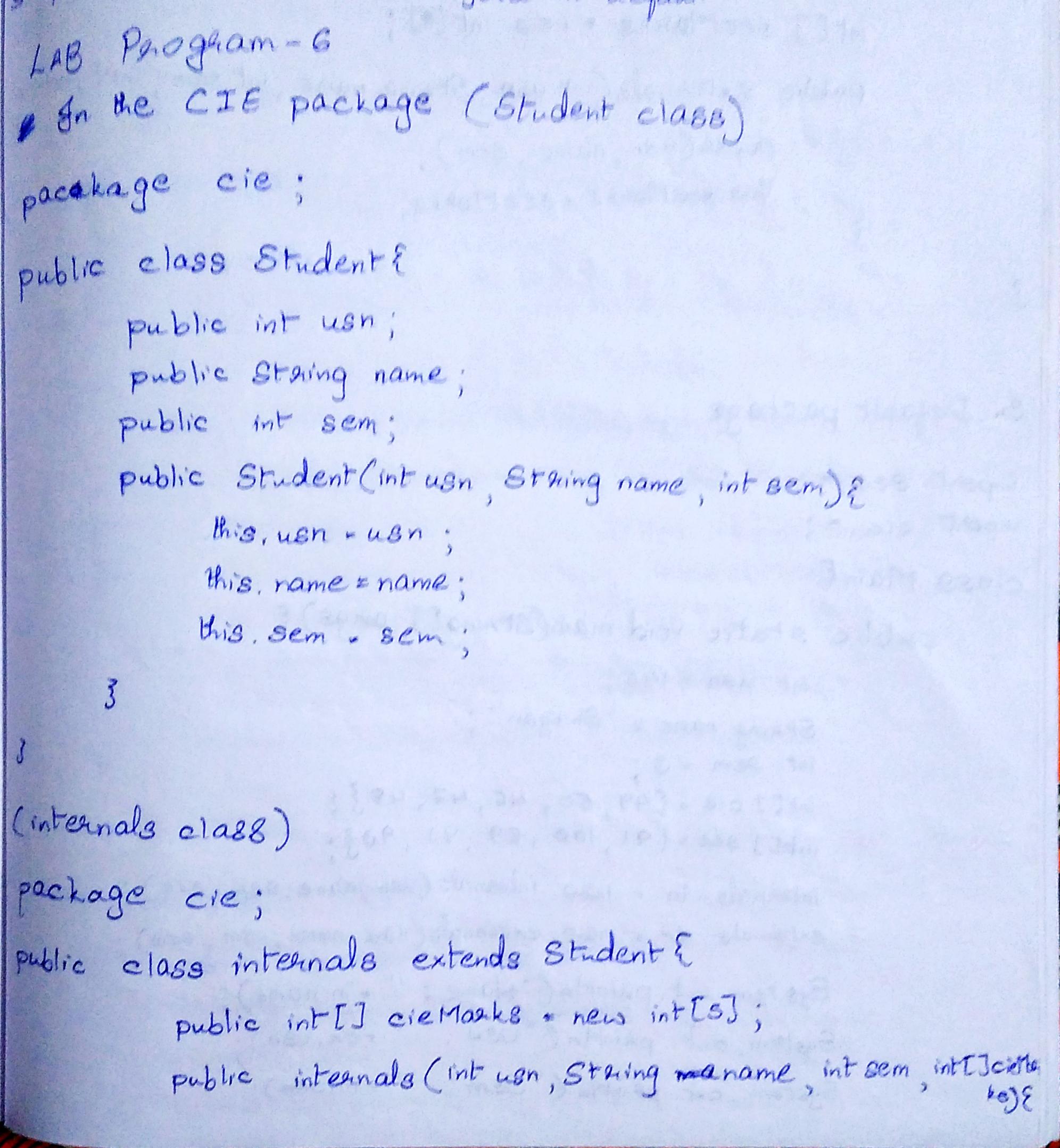
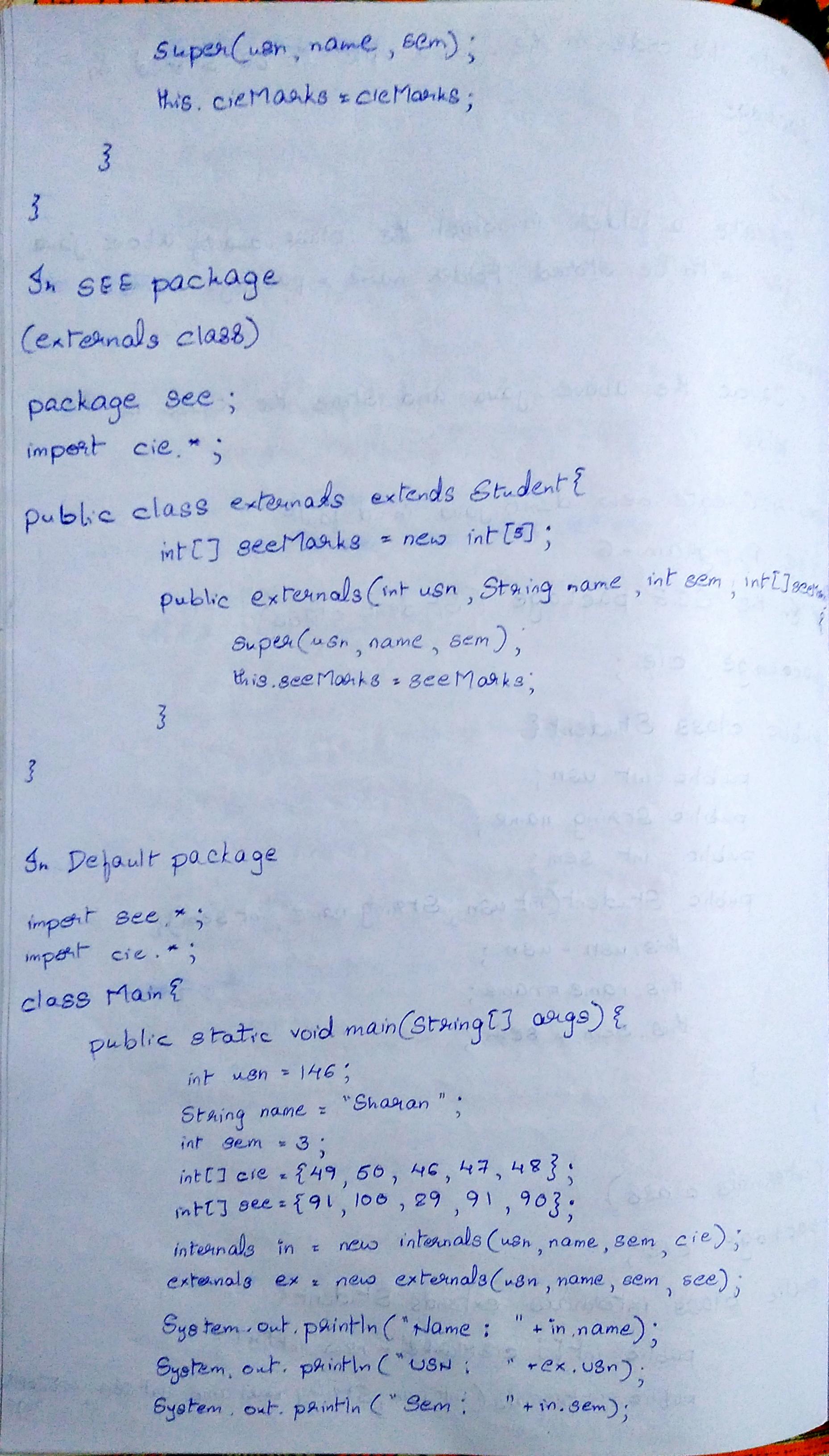
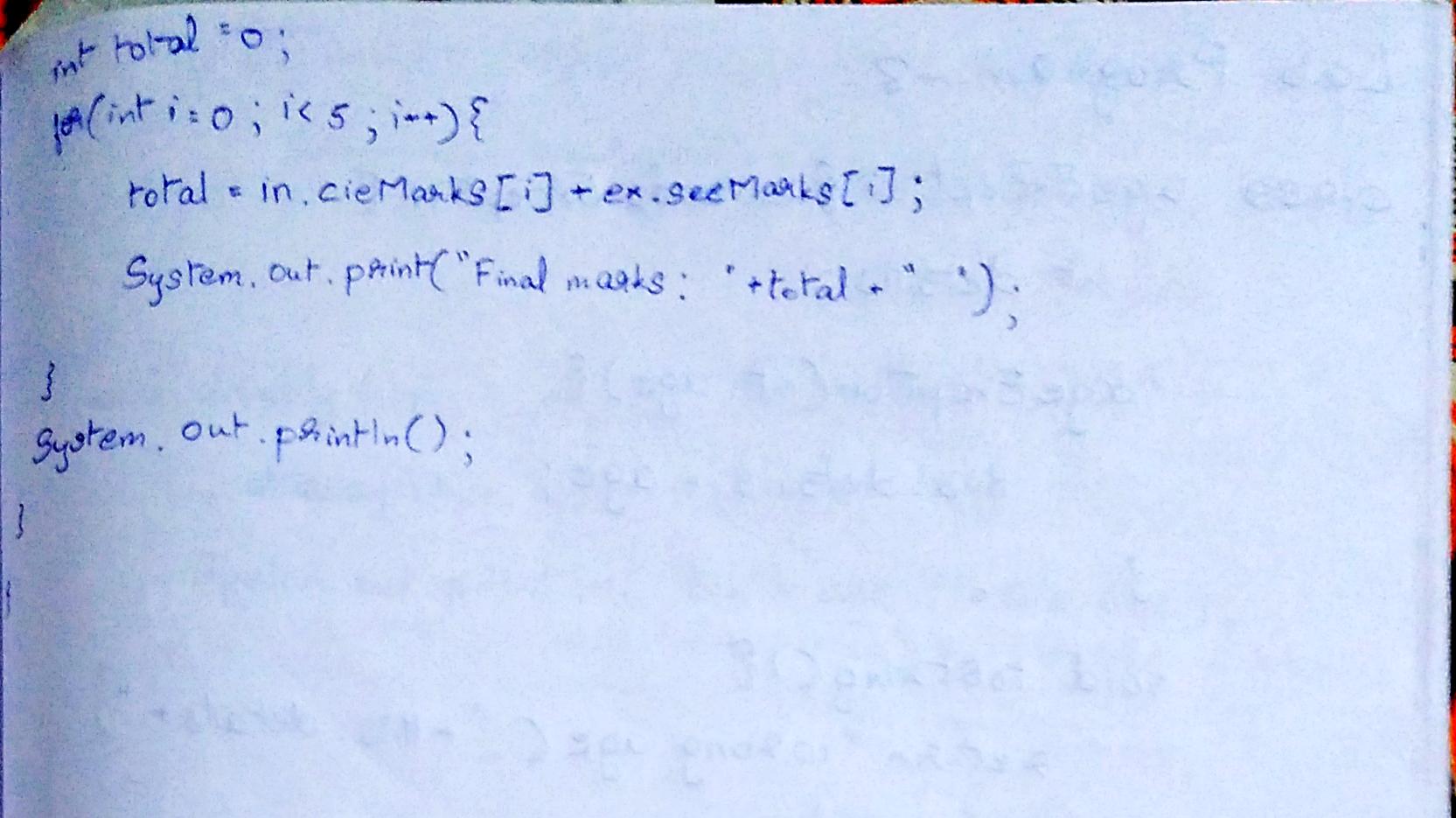
Lab Program 6

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

Write-Up







Code:

CIE package:

package cie;

public class Student{

    public int usn;

    public String name;

    public int sem;

    public Student(int usn,String name,int sem){

        this.usn = usn;

        this.name = name;

        this.sem = sem;

    }

}

package cie;

public class internals extends Student{

    public int[] cieMarks = new int[5];

    public internals(int usn,String name,int sem,int[] cieMarks){

        super(usn,name,sem);

        this.cieMarks = cieMarks;

    }

}

SEE package:

package see;

import cie.\*;

public class externals extends Student{

    public int[] seeMarks = new int[5];

    public externals(int usn,String name,int sem,int[] seeMarks){

        super(usn,name,sem);

        this.seeMarks = seeMarks;

    }

}

Main Class

import cie.\*;

import see.\*;

import java.util.\*;

class W8EP1{

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        externals[] e = new externals[2];

        internals[] in = new internals[2];

        for(int i=0;i<2;i++){

            int usn1 = input.nextInt();

            String name1 = input.next();

            int sem1 = input.nextInt();

            int[][] cie = new int[2][5];

            int[][] see = new int[2][5];

            for(int j=0;j<5;j++){

                cie[i][j] = input.nextInt();

            }

            for(int j=0;j<5;j++){

                see[i][j] = input.nextInt();

            }

            e[i] = new externals(usn1,name1,sem1,see[i]);

            in[i] = new internals(usn1,name1,sem1,cie[i]);

            int total = 0;

            System.out.println("Name: "+e[i].name);

            System.out.println("USN: "+e[i].usn);

            System.out.println("sem: "+e[i].sem);

            for(int j=0;j<5;j++){

                total = e[i].seeMarks[j]+in[i].cieMarks[j];

                System.out.print("Final marks: "+total+" ");

            }

            System.out.println();

        }

    }

}

/\*

146

sharan

3

48 50 49 48 47

100 97 92 87 91

124

karan

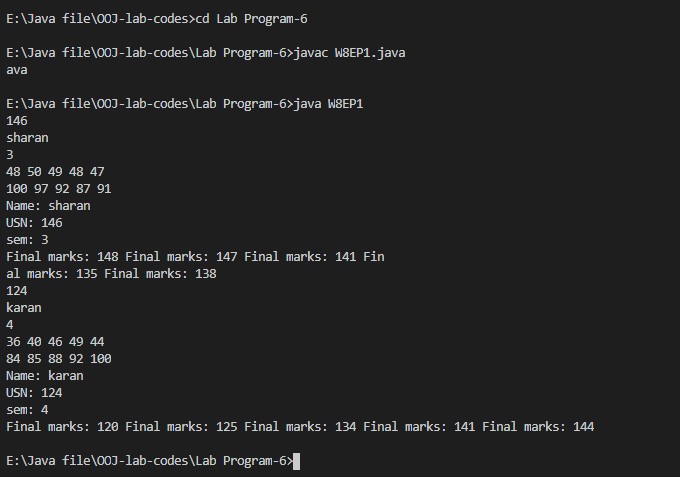
4

36 40 46 49 44

84 85 88 92 100

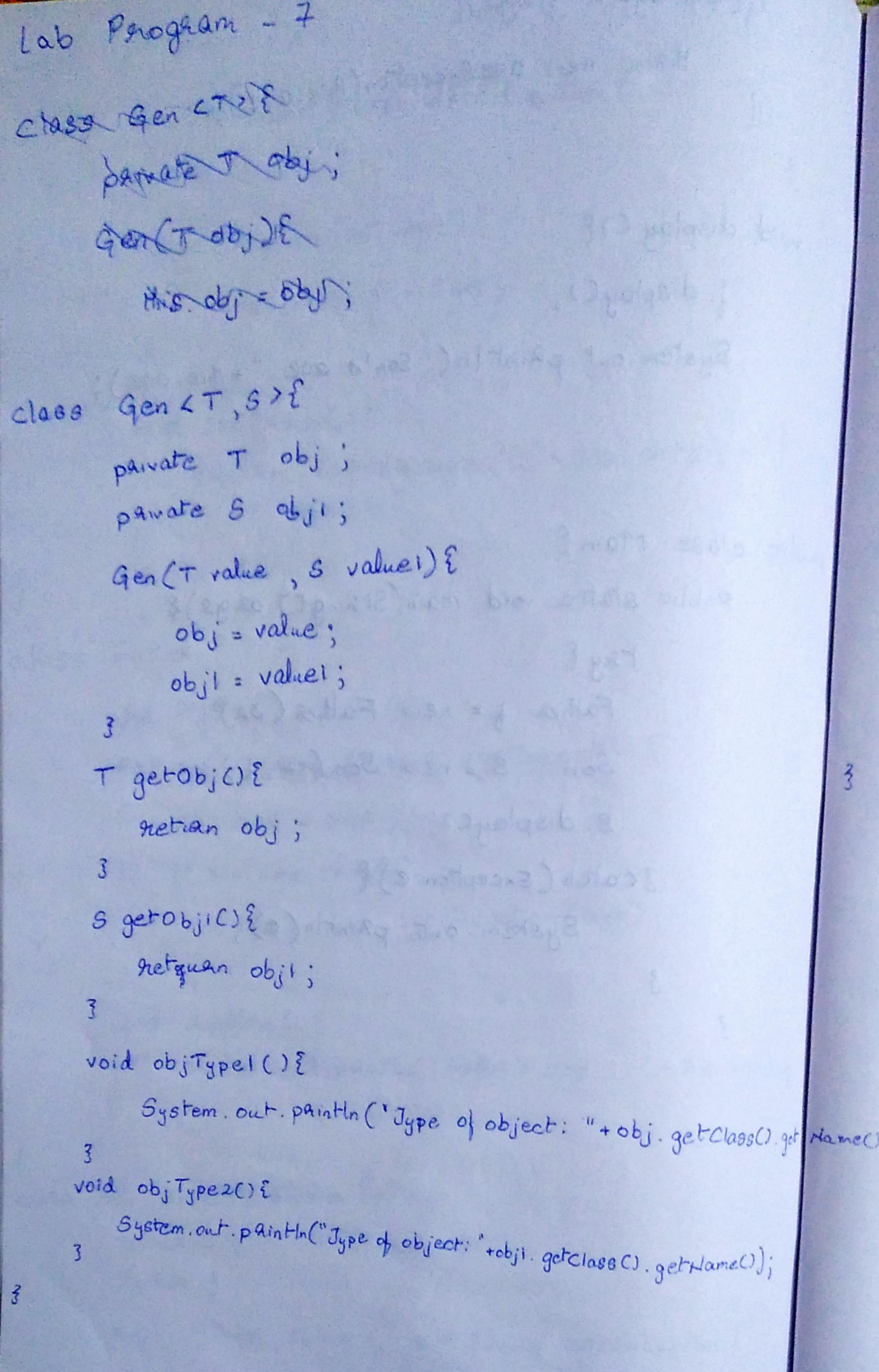
\*/

Output:



Lab Program-7

Write a program to demonstrate generics with multiple object parameters.





Code:

class Gen<T,S>{

    private T obj;

    private S obj1;

    Gen(T value,S value2){

        obj = value;

        obj1 = value2;

    }

    T getObj(){

        return obj;

    }

    S getObj1(){

        return obj1;

    }

    void objType(){

        System.out.println("The type of object "+obj.getClass().getName());

    }

    void objType1(){

        System.out.println("The type of object "+obj1.getClass().getName());

    }

}

public class Lab7{

    public static void main(String[] args){

        Gen<Integer,Double> ob = new Gen<Integer,Double>(88,88.889);

        ob.objType();

        System.out.println("Object Value "+ob.getObj());

        ob.objType1();

        System.out.println("Object Value "+ob.getObj1());

        Gen<String,Integer> ob2 = new Gen<String,Integer>("abcdefghij",12);

        ob2.objType();

        System.out.println("Object Value "+ob2.getObj());

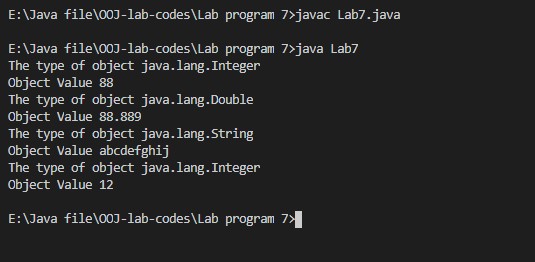
        ob2.objType1();

        System.out.println("Object Value "+ob2.getObj1());

    }

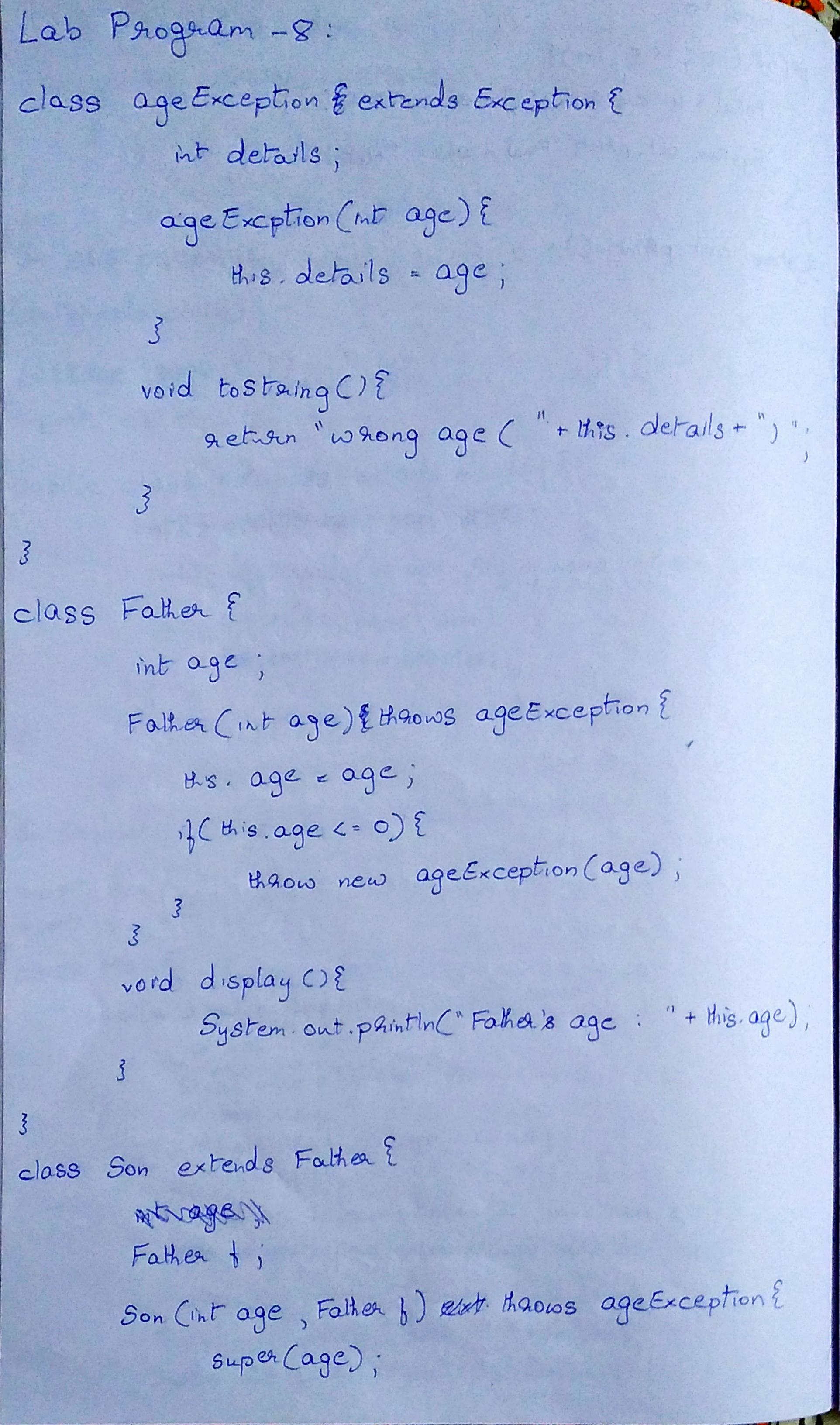
}

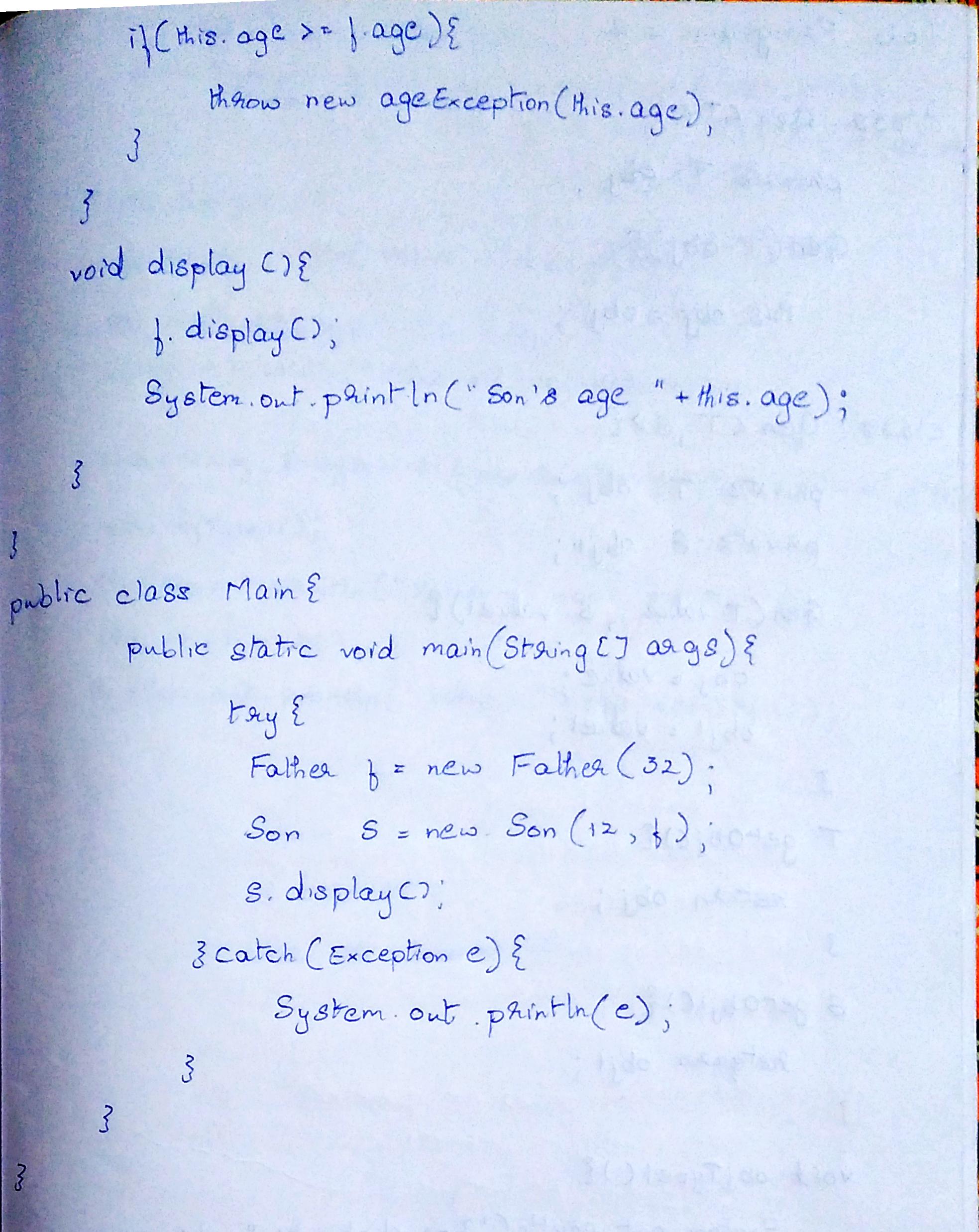
Output:



Lab Program-8

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception Wrong Age( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.





Code:

import java.util.\*;

class ageException extends Exception{

    int detail;

    ageException(int a){

        detail = a;

    }

    public String toString(){

        return "Exception :"+detail+" the enterred age does not match the category";

    }

}

class Father{

    int age;

    Father(int age) throws ageException{

        this.age = age;

    }

    void display(){

        System.out.println("Father's age:"+this.age);

    }

}

class Son extends Father{

    Father f;

    Son(int age,Father f) throws ageException{

        super(age);

        this.f = f;

    }

    void display(){

        this.f.display();

        System.out.println("Son's age:"+this.age);

    }

}

public class lab8{

    public static void main(String[] args) throws ageException{

        Scanner input = new Scanner(System.in);

        Father f = new Father(input.nextInt());

        Son s = new Son(input.nextInt(),f);

        try{

            if(s.age>=f.age)

                throw new ageException(s.age);

            if(f.age<=0)

                throw new ageException(f.age);

            if(s.age<=0)

                throw new ageException(s.age);

            s.display();

        }catch(Exception e){

            System.out.println(e);

        }

    }

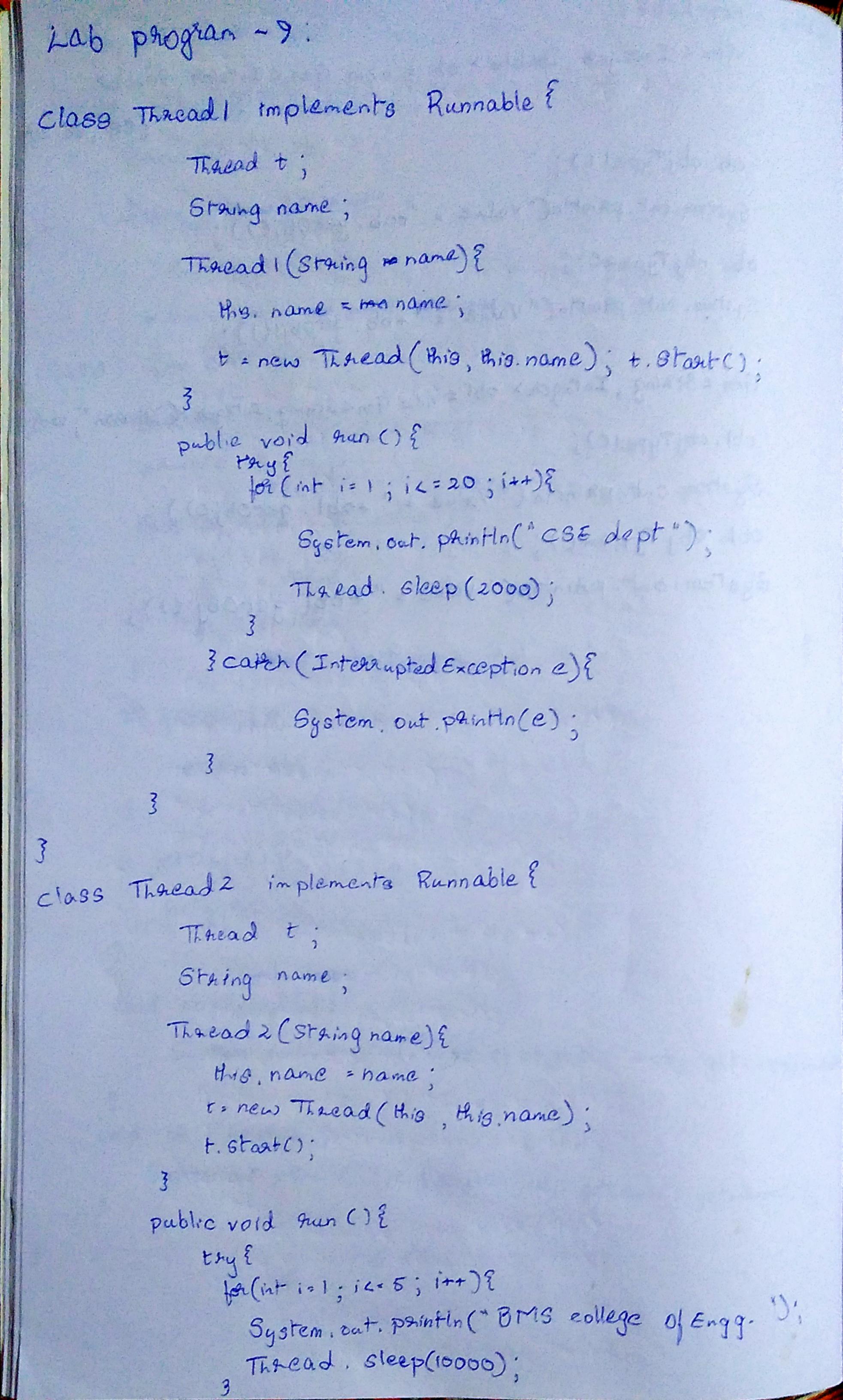
}

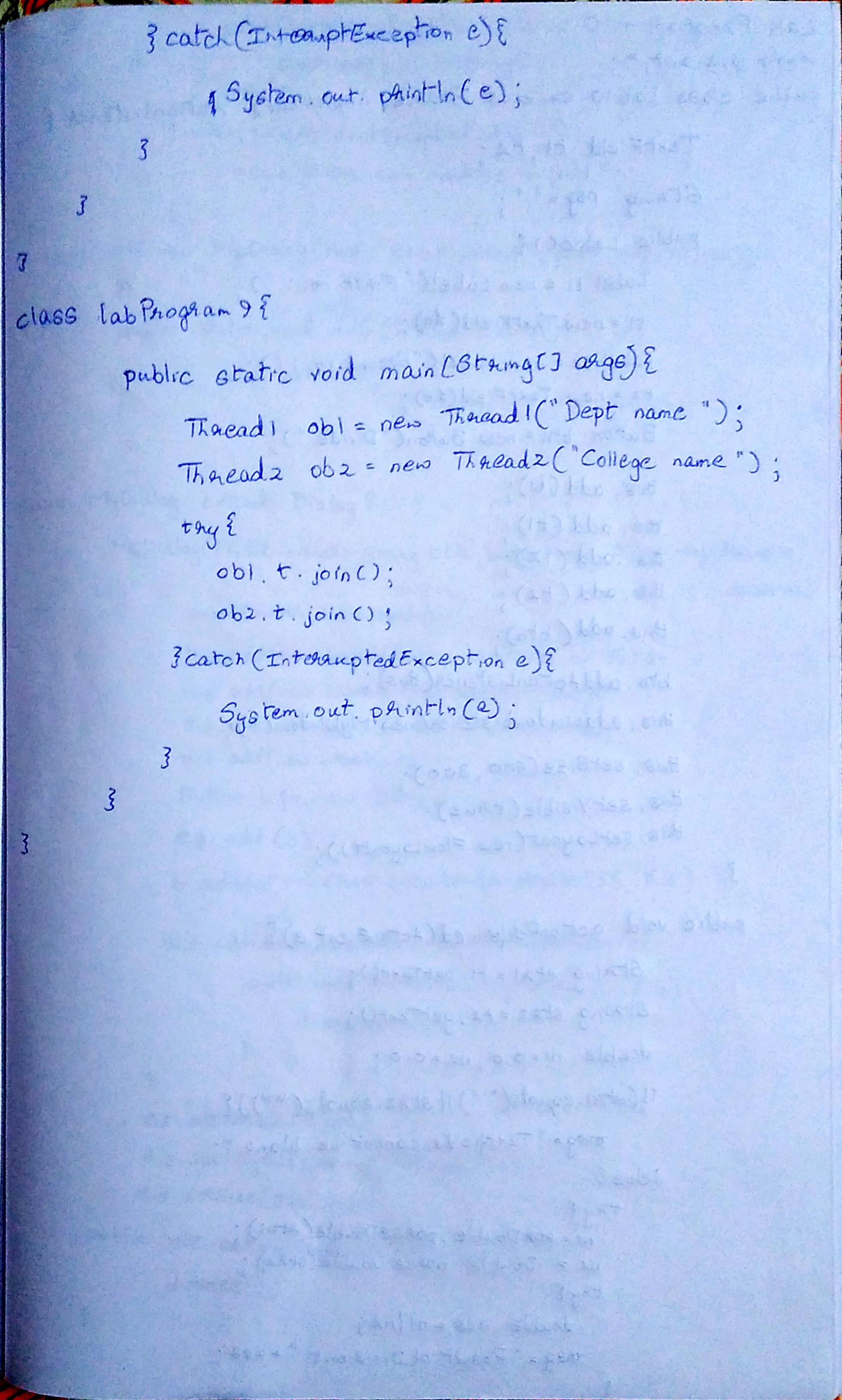
Output:



Lab Program-9

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.





Code:

class Thread1 implements Runnable{

    Thread t;

    String name;

    Thread1(String name){

        this.name = name;

        t = new Thread(this,this.name);

        t.start();

    }

    public void run(){

        try{

            for(int i=0;i<20;i++){

                System.out.println("CSE dept");

                Thread.sleep(2000);

            }

        }catch(InterruptedException e){

            System.out.println(e);

        }

    }

}

class Thread2 implements Runnable{

    Thread t;

    String name;

    Thread2(String name){

        this.name = name;

        t = new Thread(this,this.name);

        t.start();

    }

    public void run(){

        try{

            for(int i=0;i<5;i++){

                System.out.println("BMS college of Engineering");

                Thread.sleep(10000);

            }

        }catch(InterruptedException e){

            System.out.println(e);

        }

    }

}

class labProgram9{

    public static void main(String[] args){

        Thread1 obj1 = new Thread1("Dept. name");

        Thread2 obj2 = new Thread2("College name");

        //System.out.println(obj1.name+" "+obj1.t.isAlive());

        //System.out.println(obj2.name+" "+obj2.t.isAlive());

        try{

            obj1.t.join();

            obj2.t.join();

        }catch(Exception e){

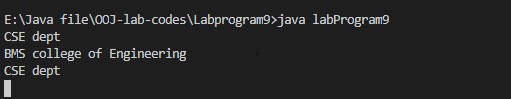
            System.out.println("Interrupted");

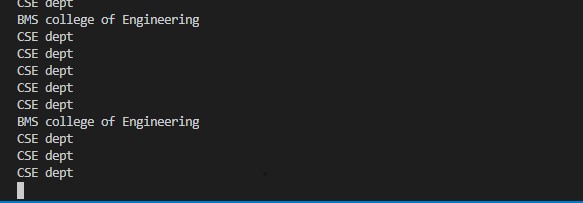
        }

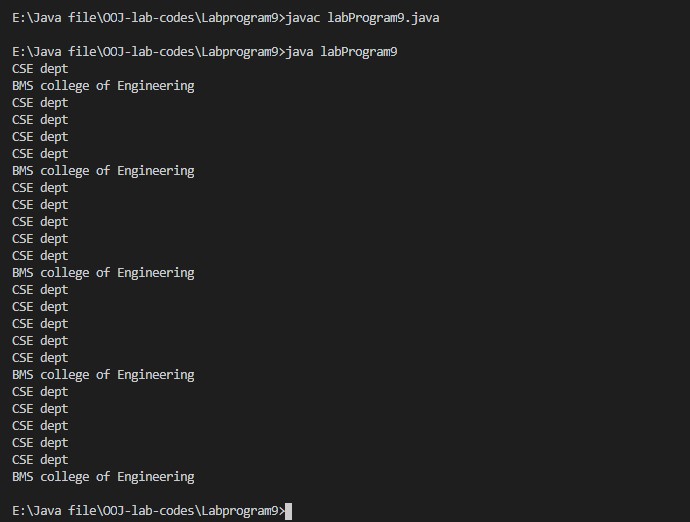
    }

}

Output:

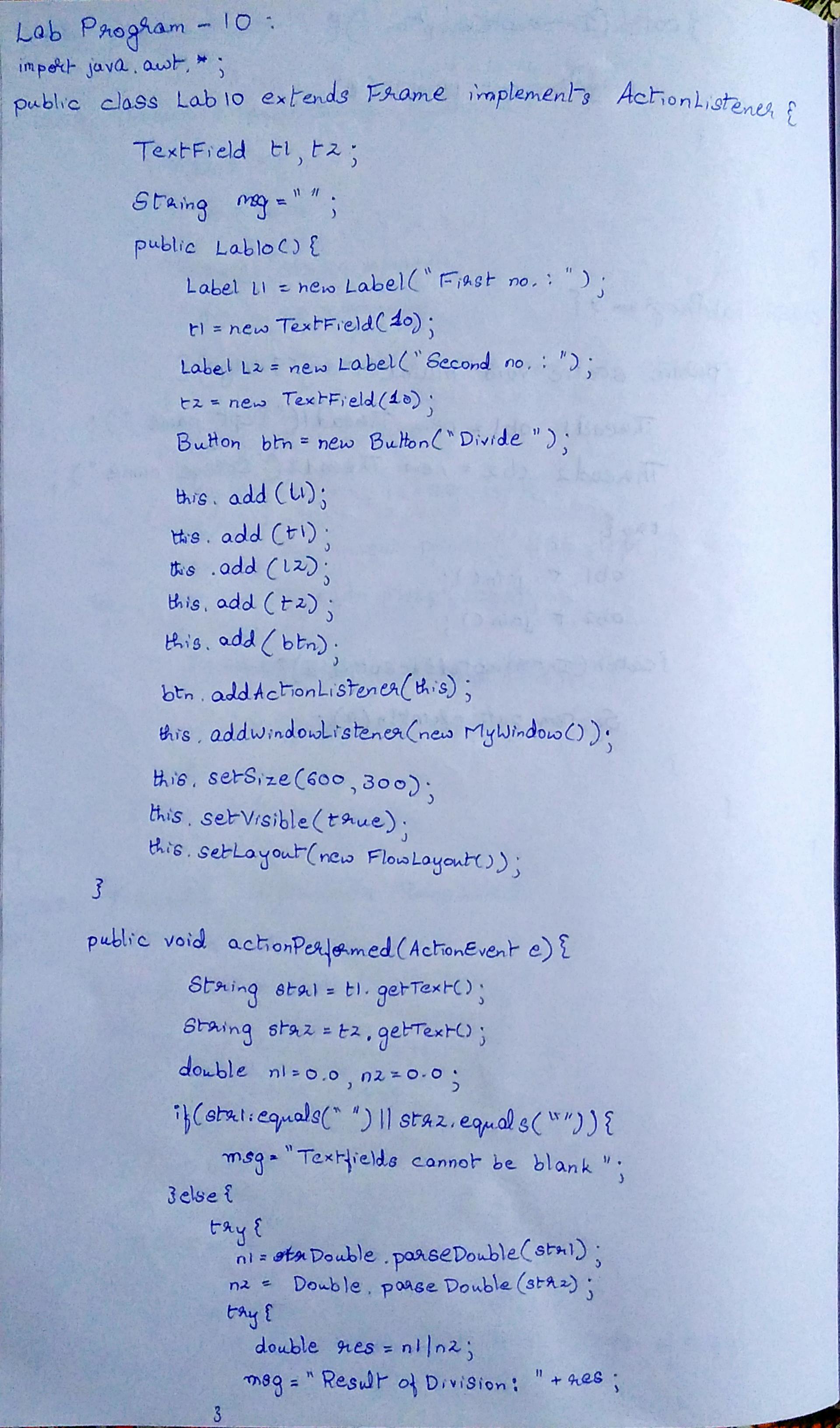


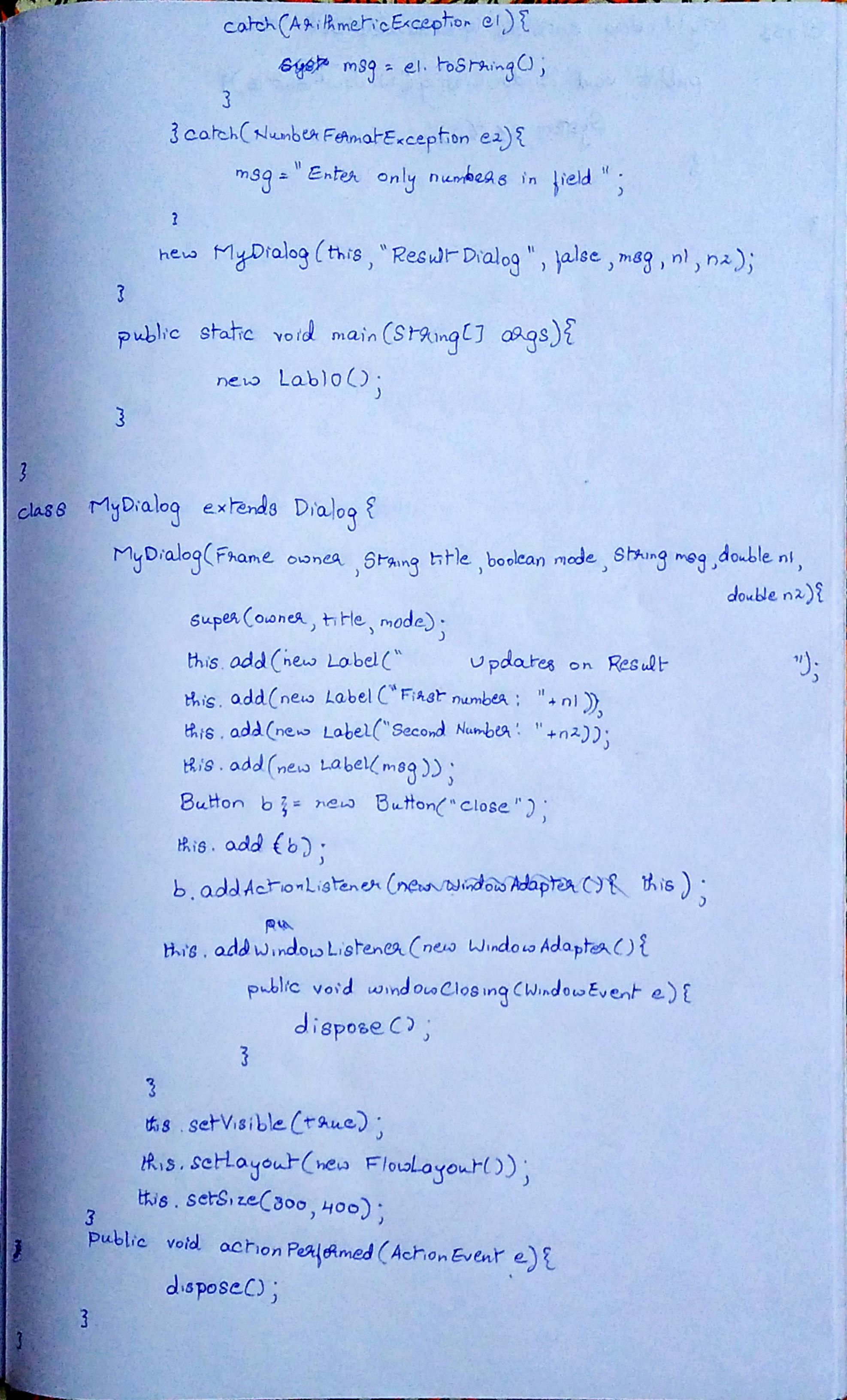


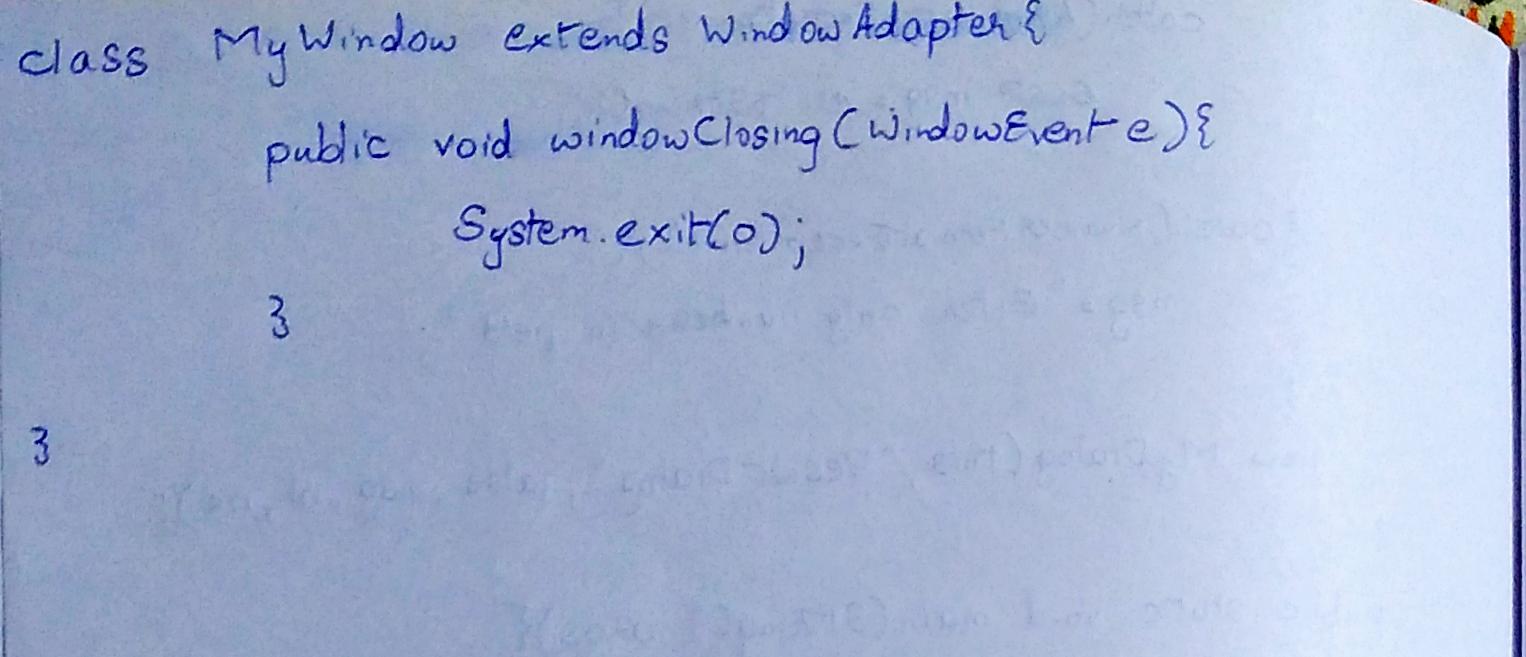


Lab Program-10

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.







Code:

import java.awt.BorderLayout;

import java.awt.Button;

import java.awt.Color;

import java.awt.Dialog;

import java.awt.FlowLayout;

import java.awt.Frame;

import java.awt.Graphics;

import java.awt.Insets;

import java.awt.Label;

import java.awt.TextField;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.TextEvent;

import java.awt.event.TextListener;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

public class Lab10 extends Frame implements ActionListener{

    TextField t1,t2;

    String msg="";

    Button btn;

    Lab10(){

        Label l1 = new Label("First Number: ",Label.RIGHT);

        t1 = new TextField(10);

        Label l2 = new Label("Second Number: ",Label.RIGHT);

        t2 = new TextField(10);

        btn = new Button("Submit");

        //Label l = new Label("Updates:");

        l1.setBackground(Color.YELLOW);

        l2.setBackground(Color.YELLOW);

        //this.setResizable(false);

        this.add(l1);

        this.add(t1);

        this.add(l2);

        this.add(t2);

        //the following command will make sure that the input char is not visible to the user

        //(it has been added just to demonstrate). Can be used for passwords.

        //t1.setEchoChar('\*');

        //t2.setEchoChar('#');

        this.add(btn,BorderLayout.CENTER);

        this.setVisible(true);

        this.setSize(600, 300);

        this.setLayout(new FlowLayout(FlowLayout.CENTER,20,10));

        //t1.addActionListener(this);

        btn.addActionListener(this);

        addWindowListener(new MyWindow());

        setBackground(Color.YELLOW);

        //System.out.println(BorderLayout.CENTER);

    }

    @Override

    public Insets getInsets() {

        return new Insets(50,10,10,20);

    }

    @Override

    public void actionPerformed(ActionEvent e) {

        String st1 = t1.getText();

        String st2 = t2.getText();

        double n1,n2;

        n1 = 0.0;

        n2 = 0.0;

        if(st1.equals("")||st2.equals("")) {

            msg="You cannot leave the text elements blank";

        }else{

            try {

                n1 = Double.parseDouble(st1);

                n2 = Double.parseDouble(st2);

                try {

                    double res = n1/n2;

                    msg = "Result of division: "+res;

                }catch(ArithmeticException e1) {

                    msg = e1.toString();

                }

            }catch(NumberFormatException e2) {

                msg = "Enter only numbers and not other things";

            }

        }

        new MyDialog(this,"Result Dialog",false,msg,n1,n2);

    }

    public static void main(String[] args) {

        new Lab10();

    }

}

class MyDialog extends Dialog implements ActionListener{

    public MyDialog(Frame owner, String title, boolean modal,String msg, double n1, double n2) {

        super(owner, title, modal);

        this.setVisible(true);

        this.setSize(300, 400);

        this.setLayout(new FlowLayout());

        //System.out.println(owner);

        Label l1 = new Label("          Updates on the result:          ");

        //l1.setSize(300, 20);

        this.add(l1);

        this.add(new Label("First Number: "+n1));

        this.add(new Label("Second Number: "+n2));

        this.add(new Label(msg));

        Button b = new Button("Close");

        this.add(b);

        b.addActionListener(this);

        this.addWindowListener(new WindowAdapter() {

            public void windowClosing(WindowEvent e) {

                dispose();

            }

        });

    }

    @Override

    public void actionPerformed(ActionEvent e) {

        dispose();

    }

}

class MyWindow extends WindowAdapter{

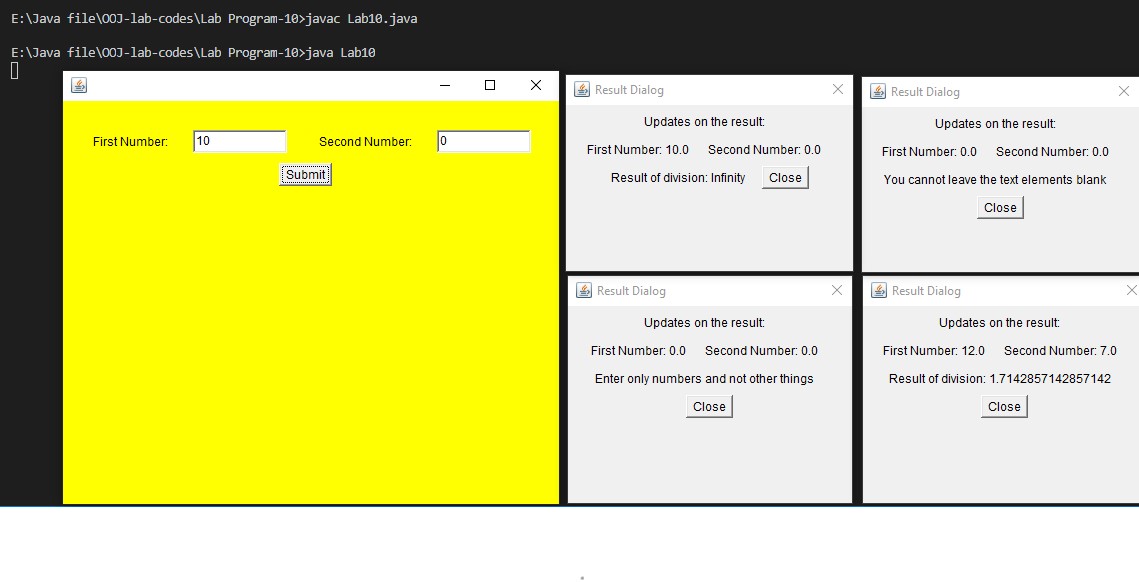
    public void windowClosing(WindowEvent e) {

        System.exit(0);

    }

}

Output:



Lab Record:

Sharan.S.Pai

1BM19CS146,

BMS College Of Engineering.

Lab In-charge:

**Pallavi G.B.,**

Assistant Professor, CSE Department,

BMS College of Engineering, Basavanagudi, Bangalore-560019

Ph. No: +91-9886664817