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

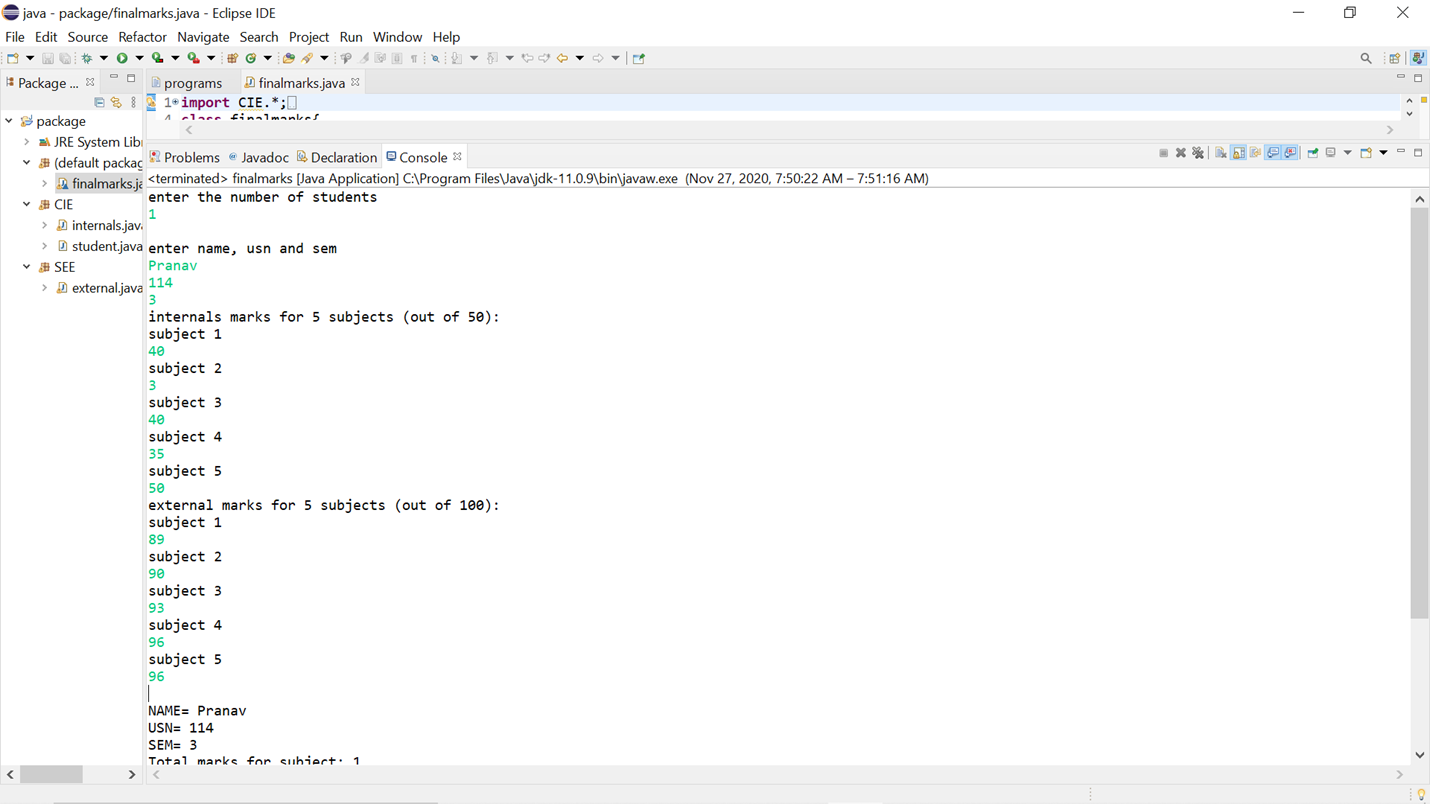
semester of the student. Create another package SEE which has the class

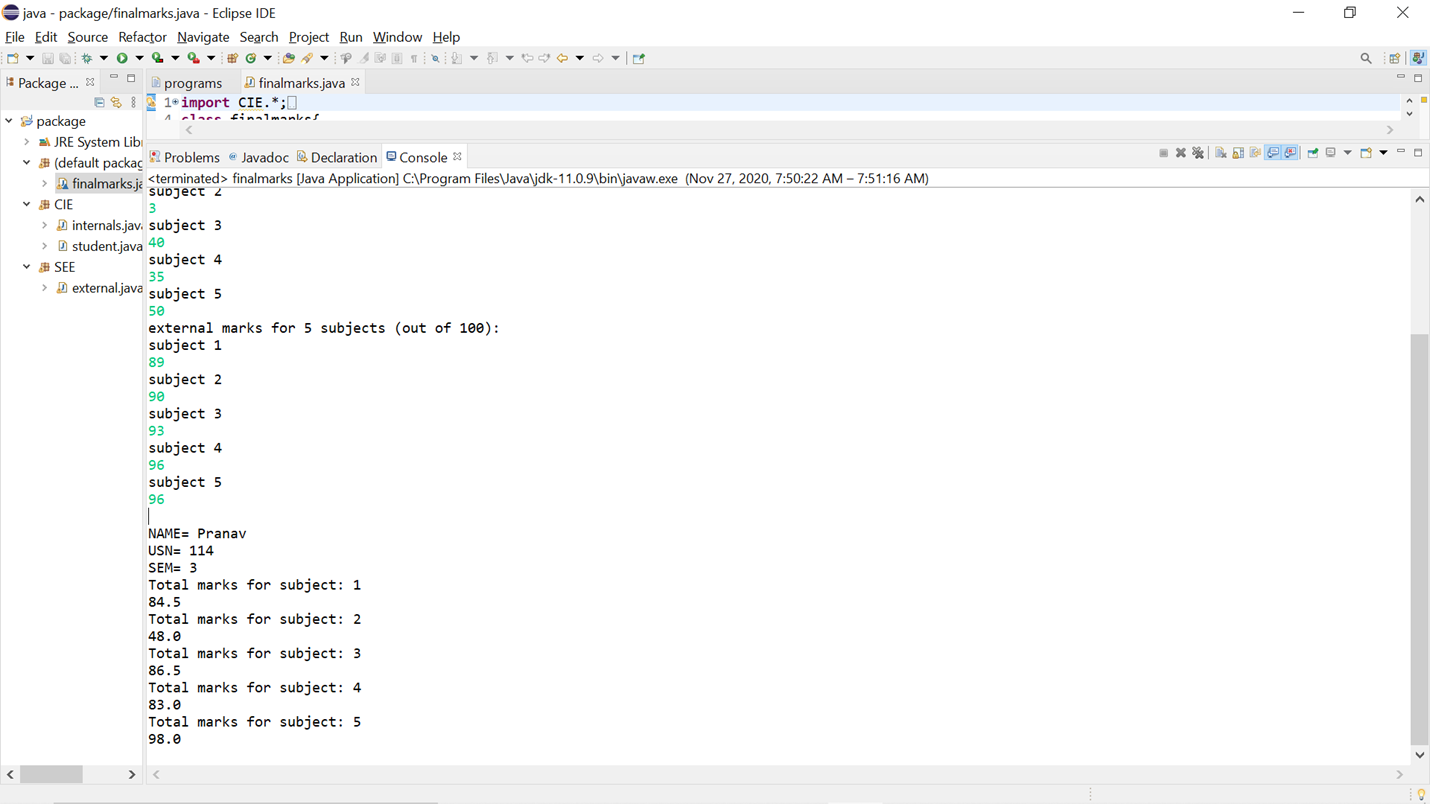
External which is a derived class of Student. This class has an array that

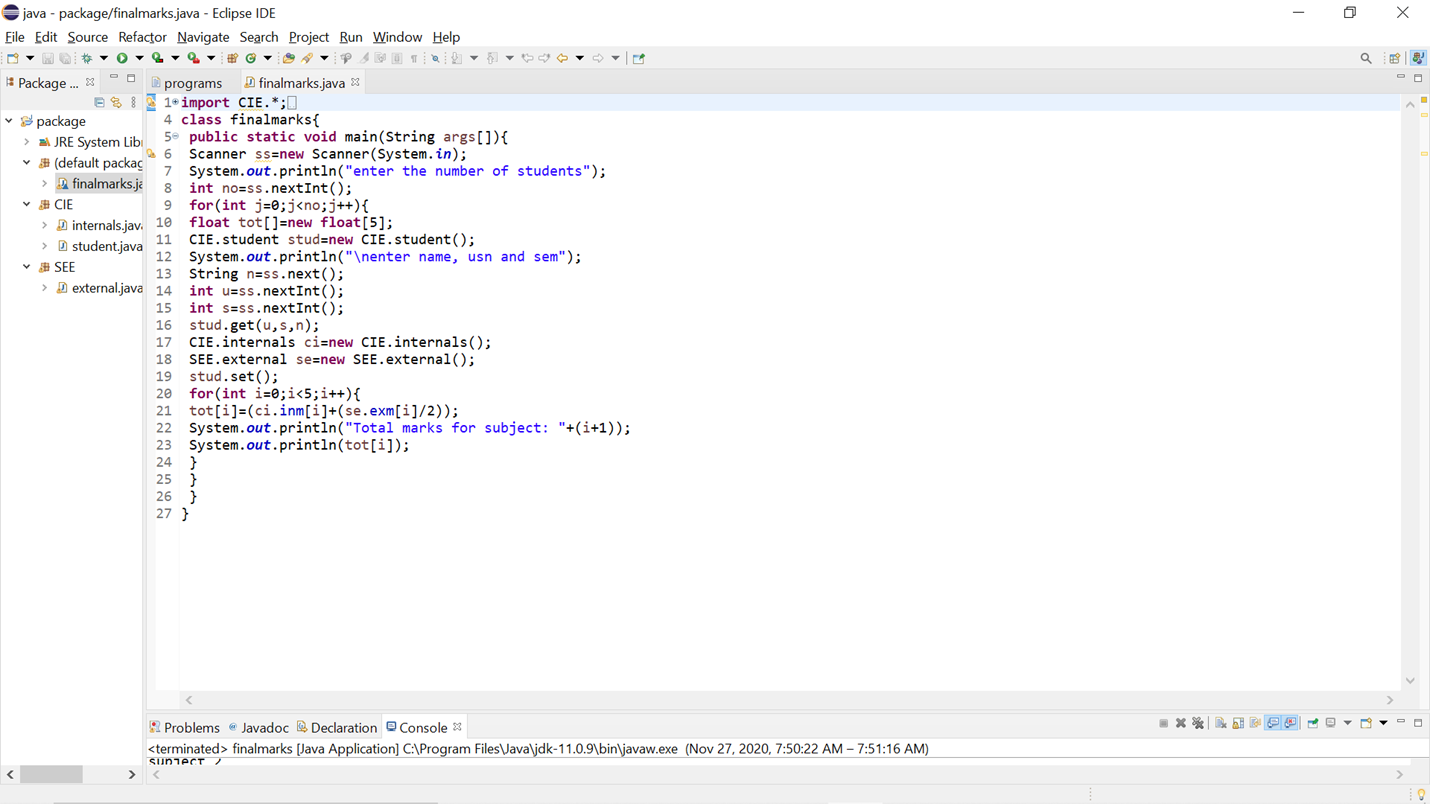
stores the SEE marks scored in five courses of the current semester of the

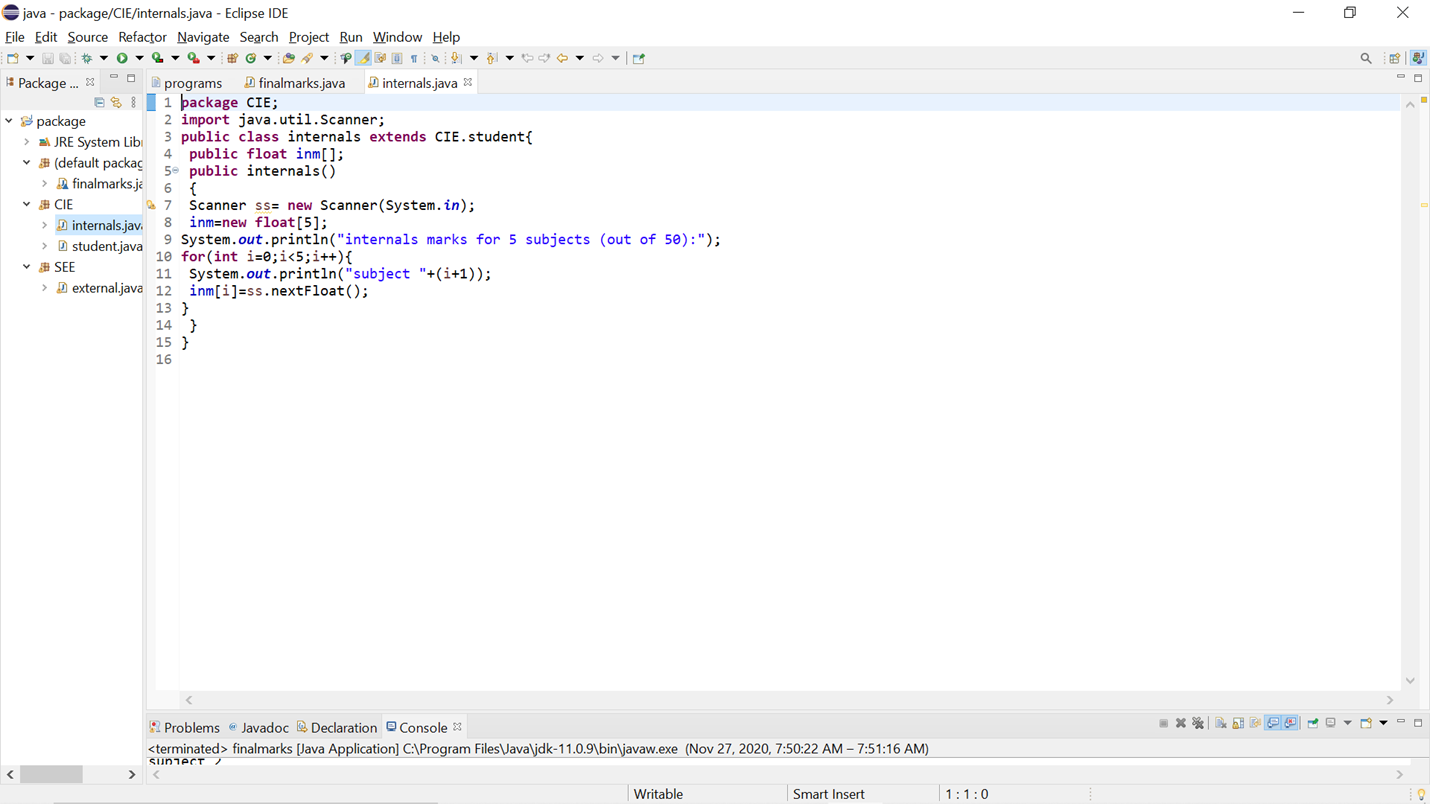
student. Import the two packages in a file that declares the final marks of n

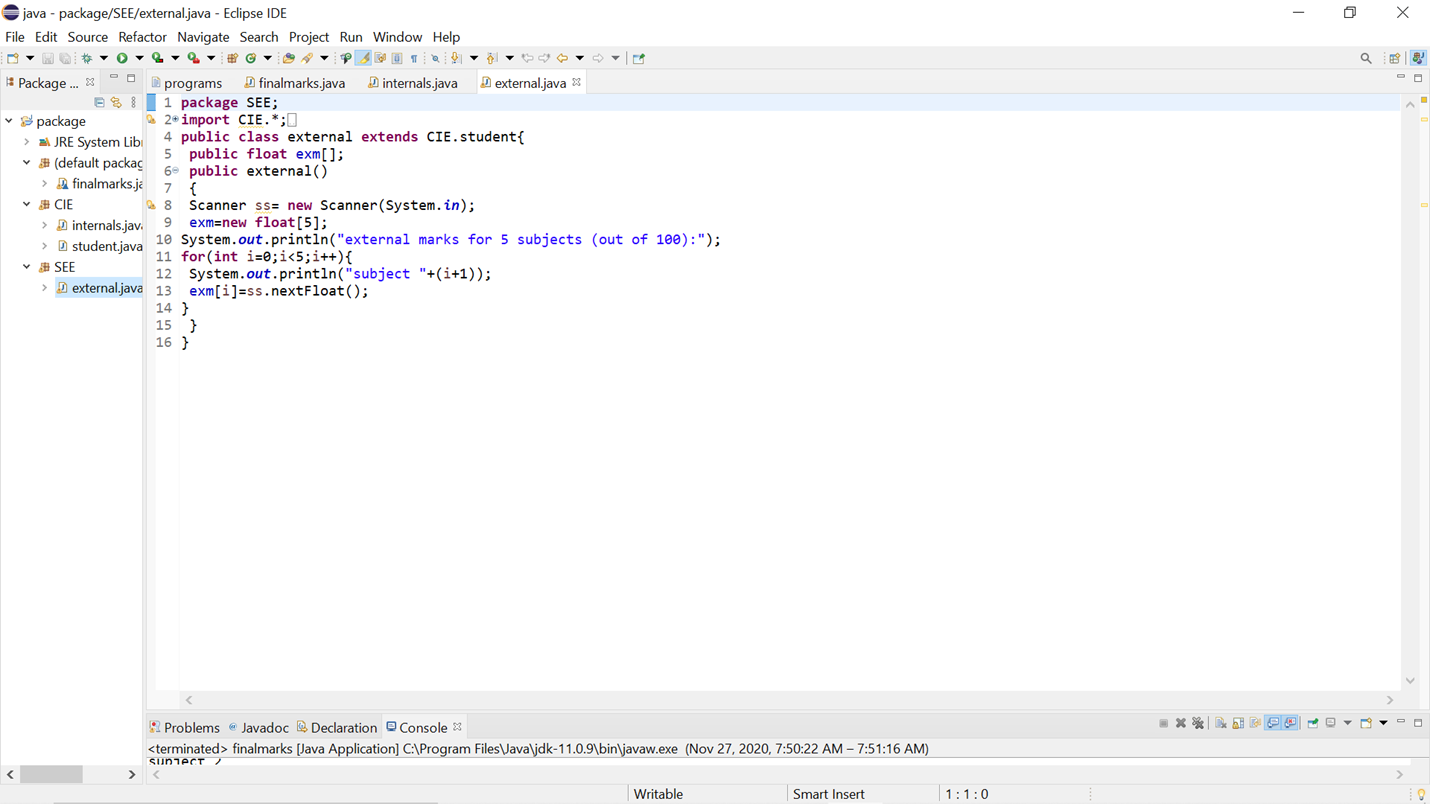
students in all five courses.





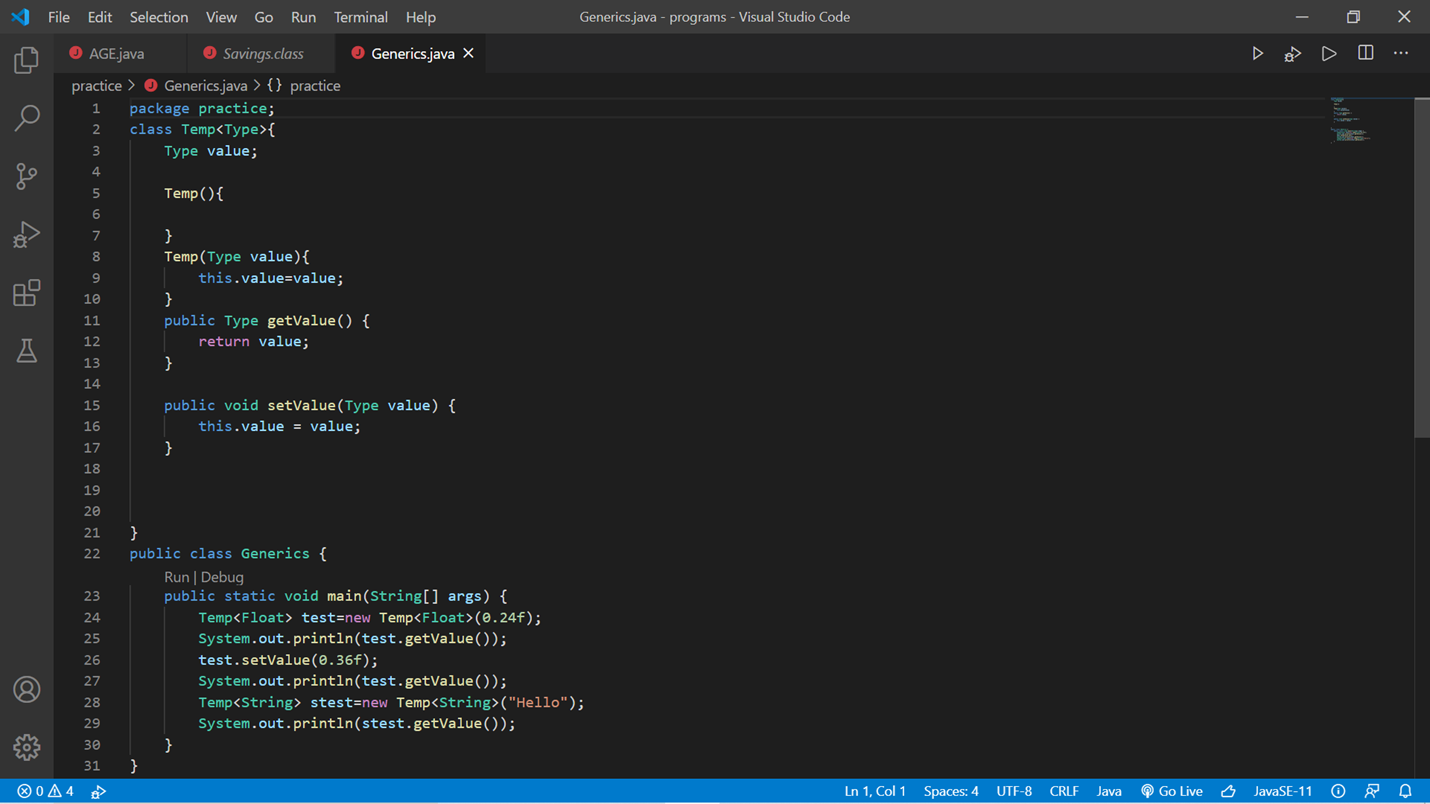


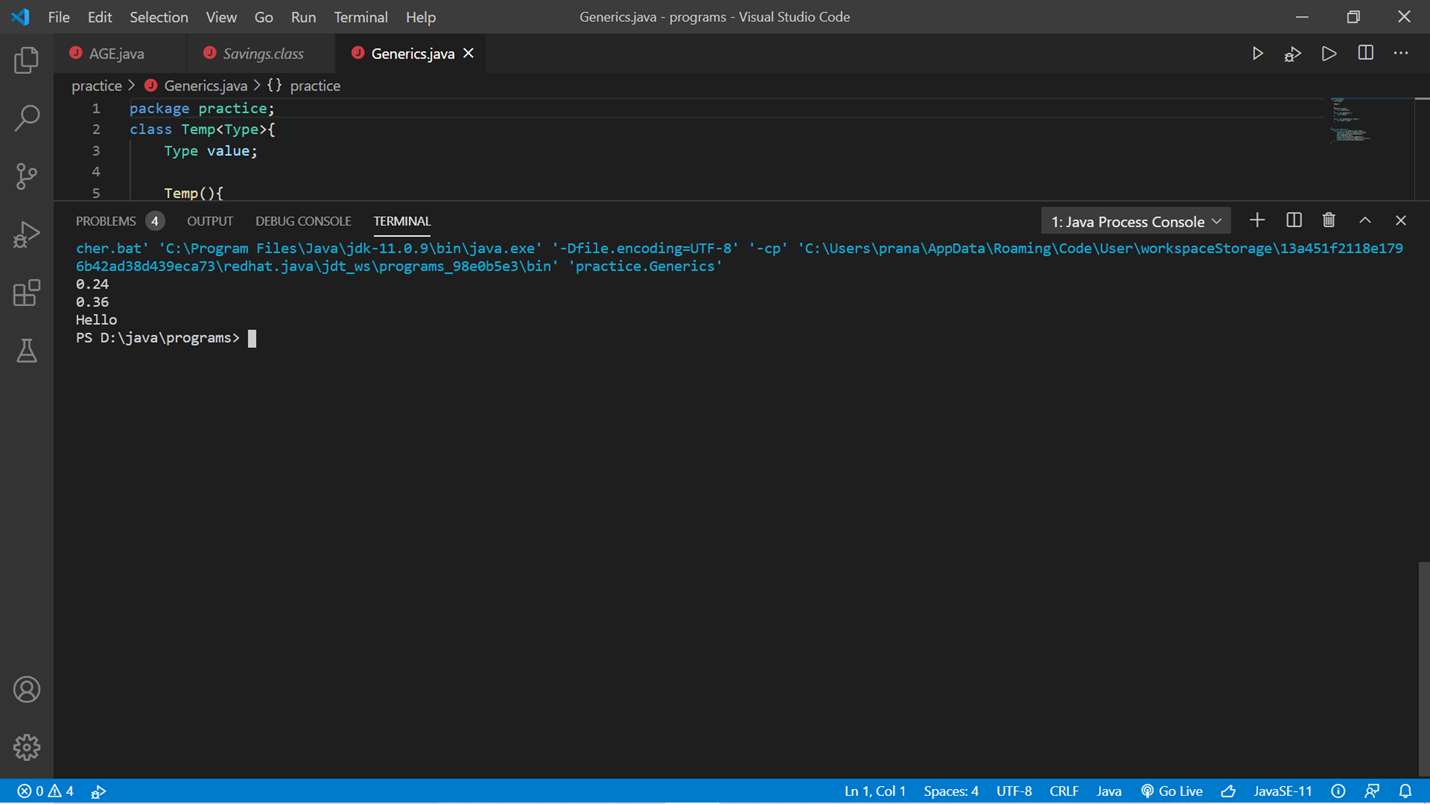




Program 7

1. Write a program to demonstrate generics with multiple object parameters.





class Temp<Type>{

    Type value;

    Temp(){

    }

    Temp(Type value){

        this.value=value;

    }

    public Type getValue() {

        return value;

    }

    public void setValue(Type value) {

        this.value = value;

    }

}

public class lab\_program\_7 {

    public static void main(String[] args) {

        Temp<Float> test=new Temp<Float>(0.24f);

        System.out.println(test.getValue());

        test.setValue(0.36f);

        System.out.println(test.getValue());

        Temp<String> stest=new Temp<String>("Hello");

        System.out.println(stest.getValue());

    }

}

Program 8

2. Write a program that demonstrates handling of exceptions in inheritance tree. Create a

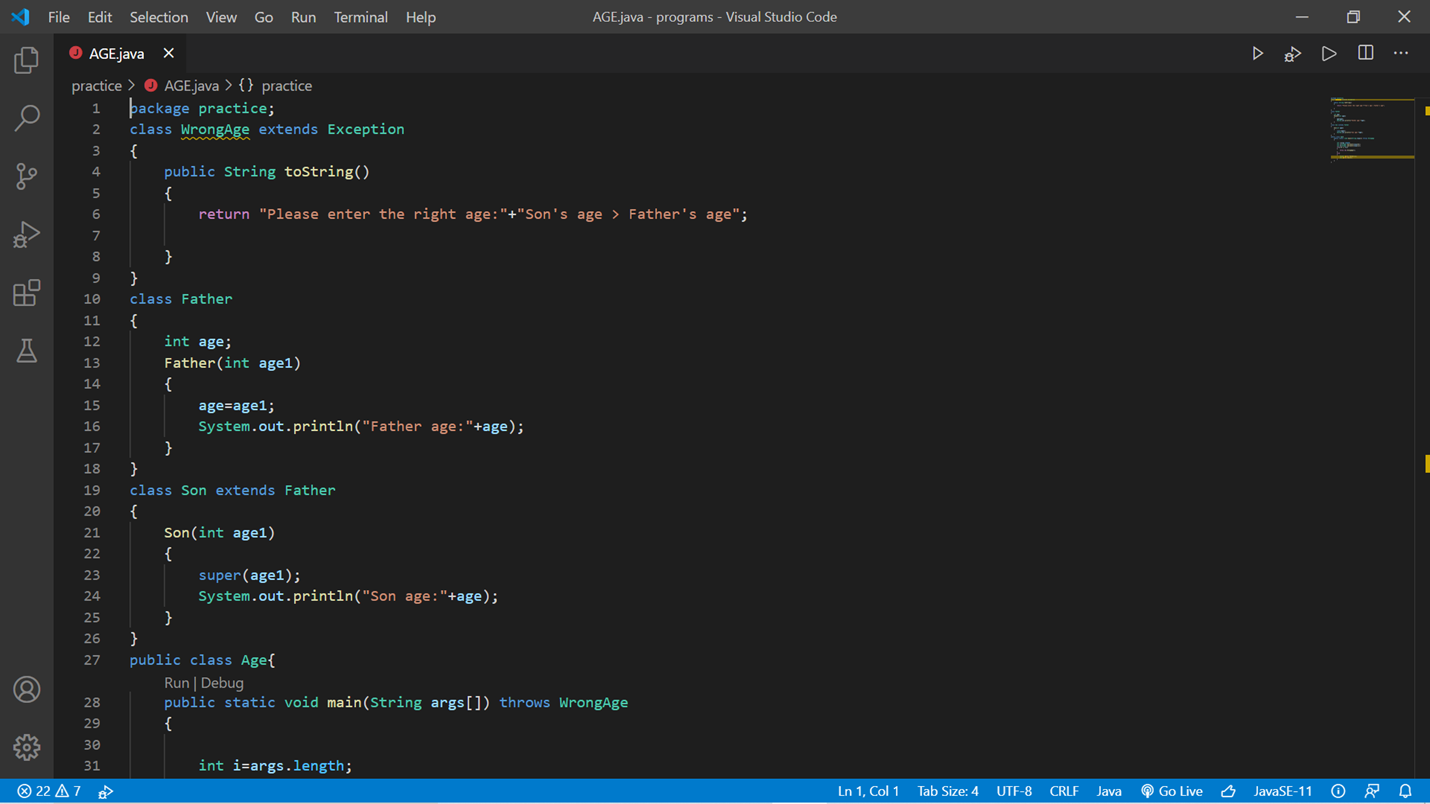
base class called “Father” and derived class called “Son” which extends the base

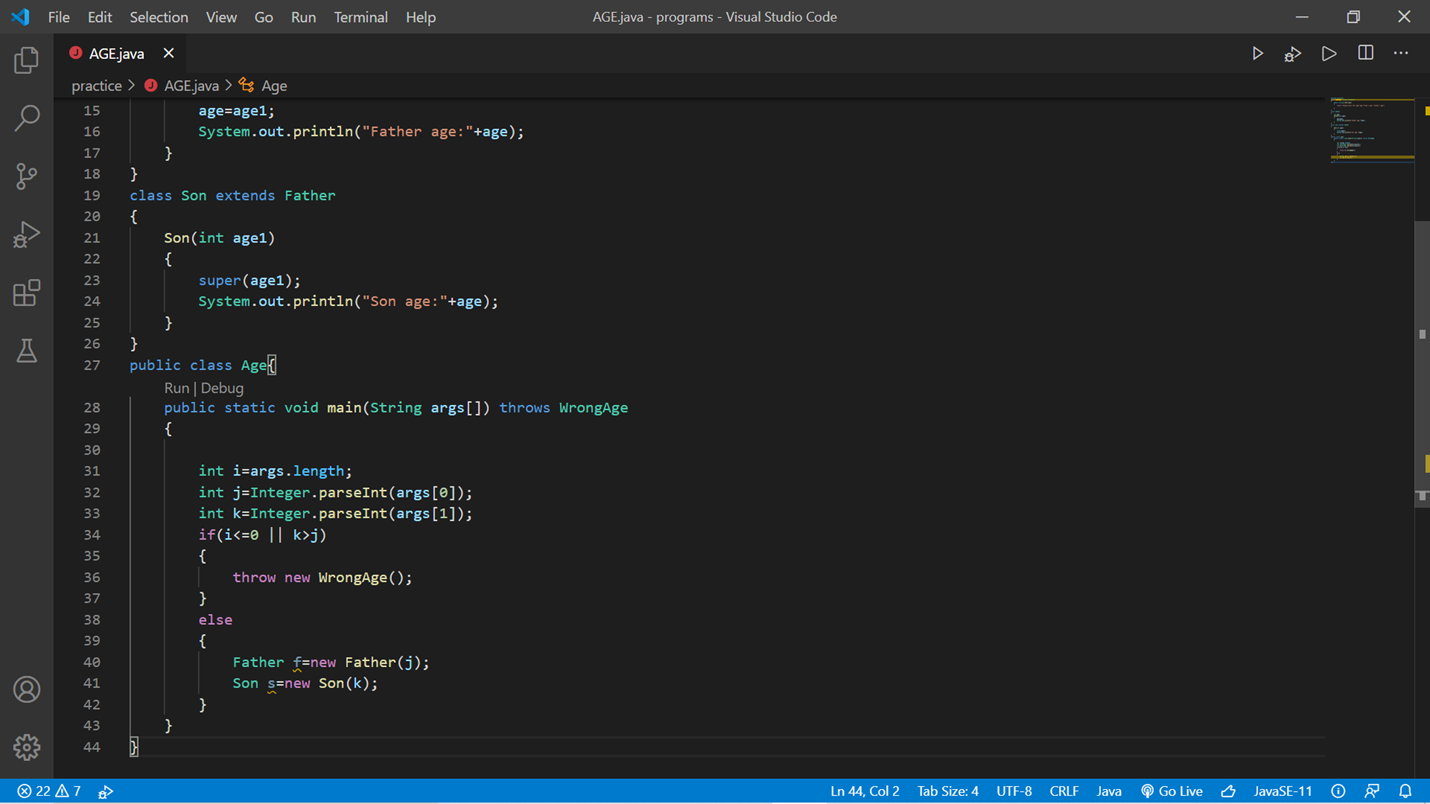
class. In Father class, implement a constructor which takes the age and throws the

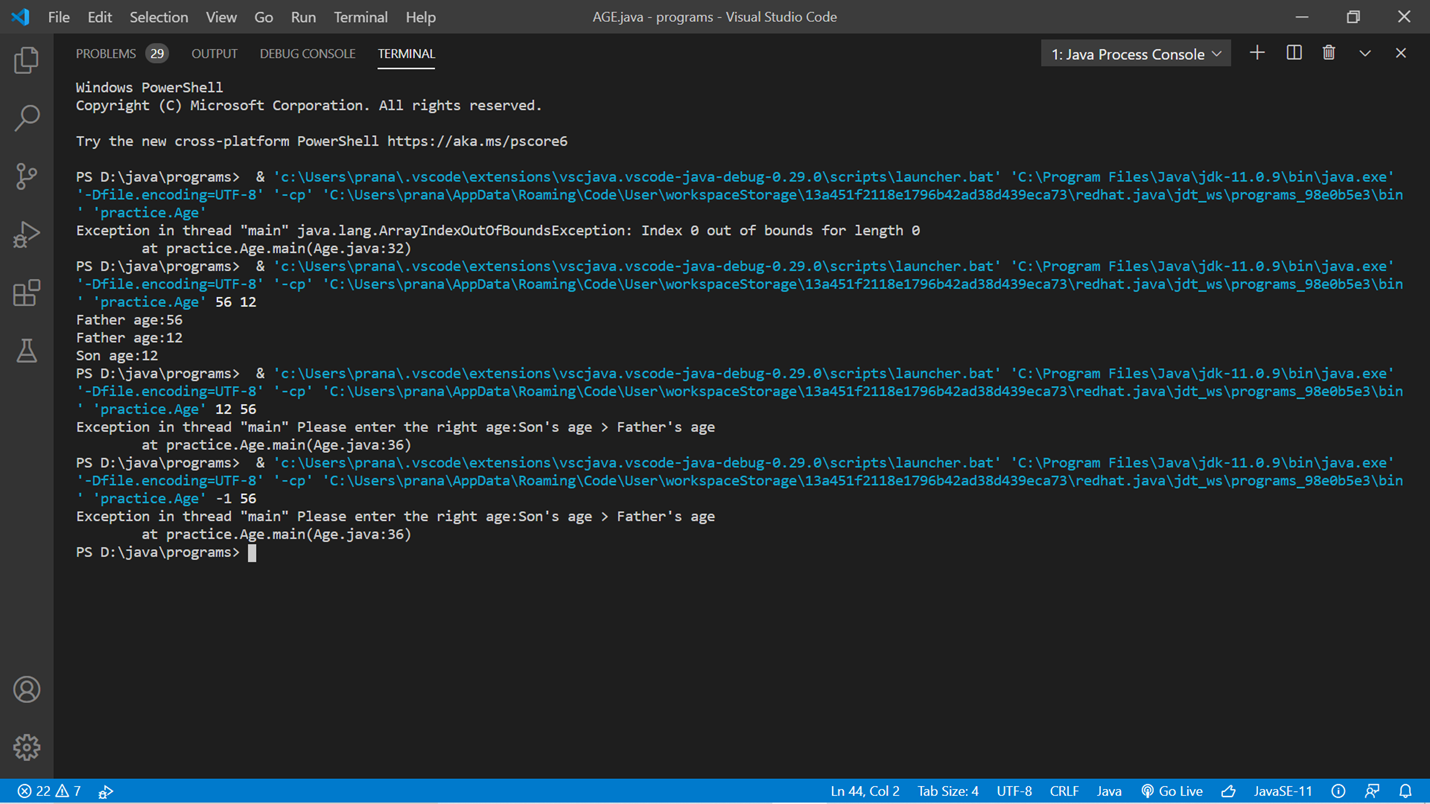
exception WrongAge( ) when the input age&lt;0. In Son class, implement a constructor

that cases both father and son’s age and throws an exception if son’s age is &gt;=father’s

age.







package practice;

class WrongAge extends RuntimeException

{

    WrongAge(String msg){

        super(msg);

    }

}

class Father

{

    int age;

    Father(int age1)

    {

        age=age1;

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

    }

}

class Son extends Father

{

    Son(int age1)

    {

        super(age1);

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

    }

}

public class Age{

    public static void main(String args[]) throws WrongAge

    {

    int i=args.length;

        int j=Integer.parseInt(args[0]);

        int k=Integer.parseInt(args[1]);

        try {

            if(i<=0 || k>j)

            {

                throw new WrongAge("Son age can't be greater than Father"); //this will create the exception object and the jvm will search if the exception is handled or not

            }

            else

            {

                new Father(j);

                new Son(k);

            }

        }

        catch (WrongAge e) {

            // e.getMessage() //print only the message

            // e.toString()

            e.printStackTrace(); //this is done to print the message ,description ,and stacktrace of the error

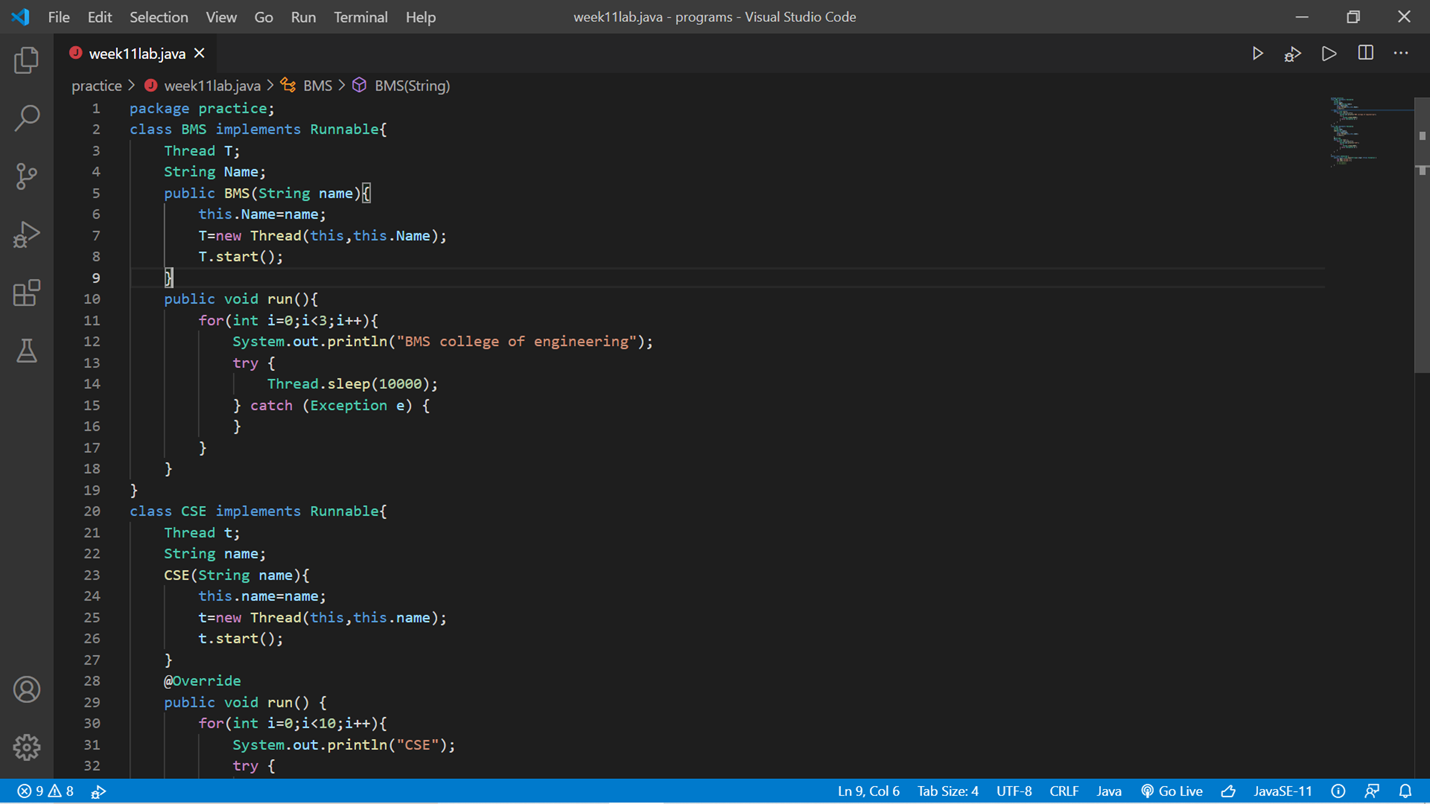
        }

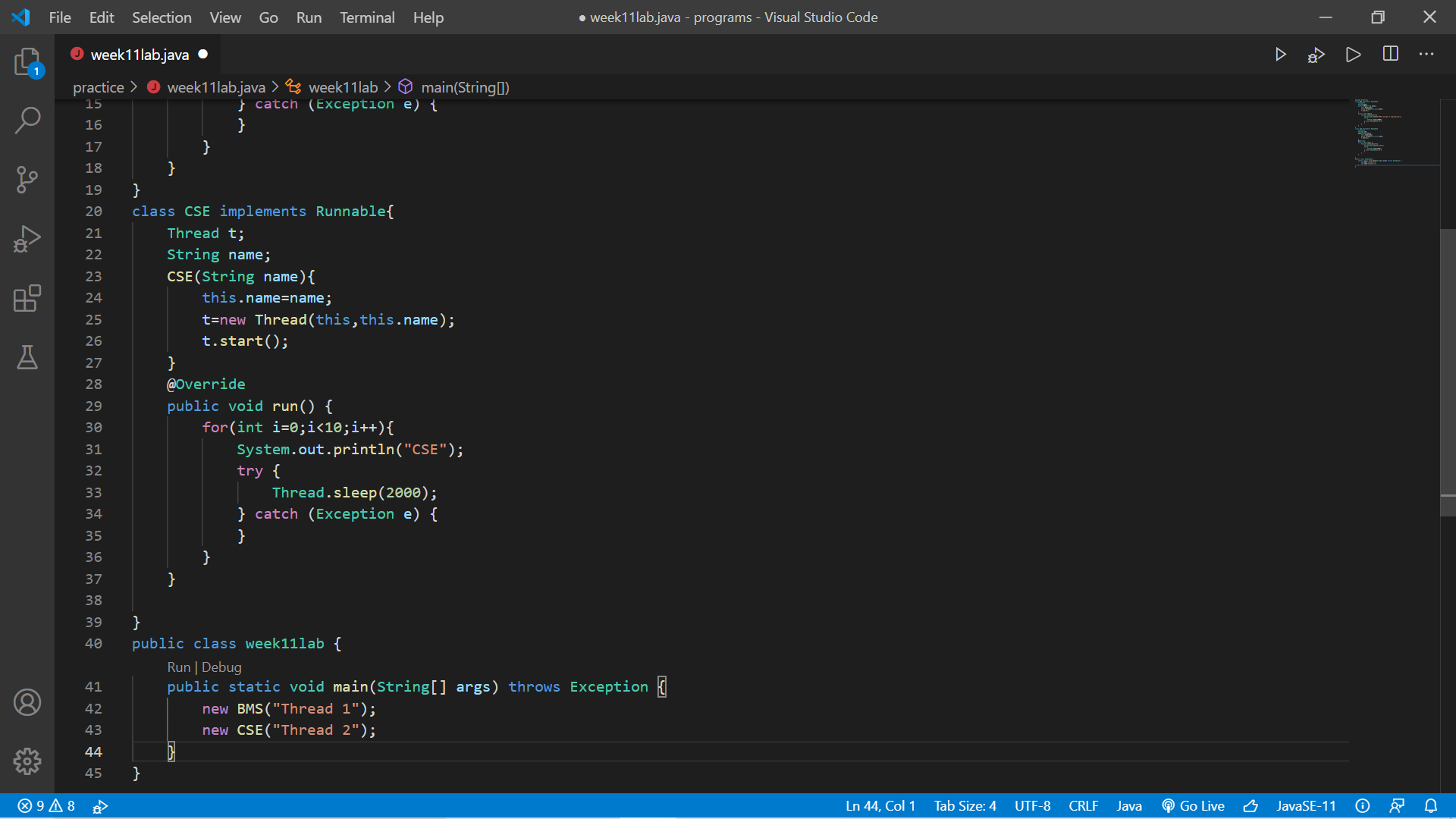
    }

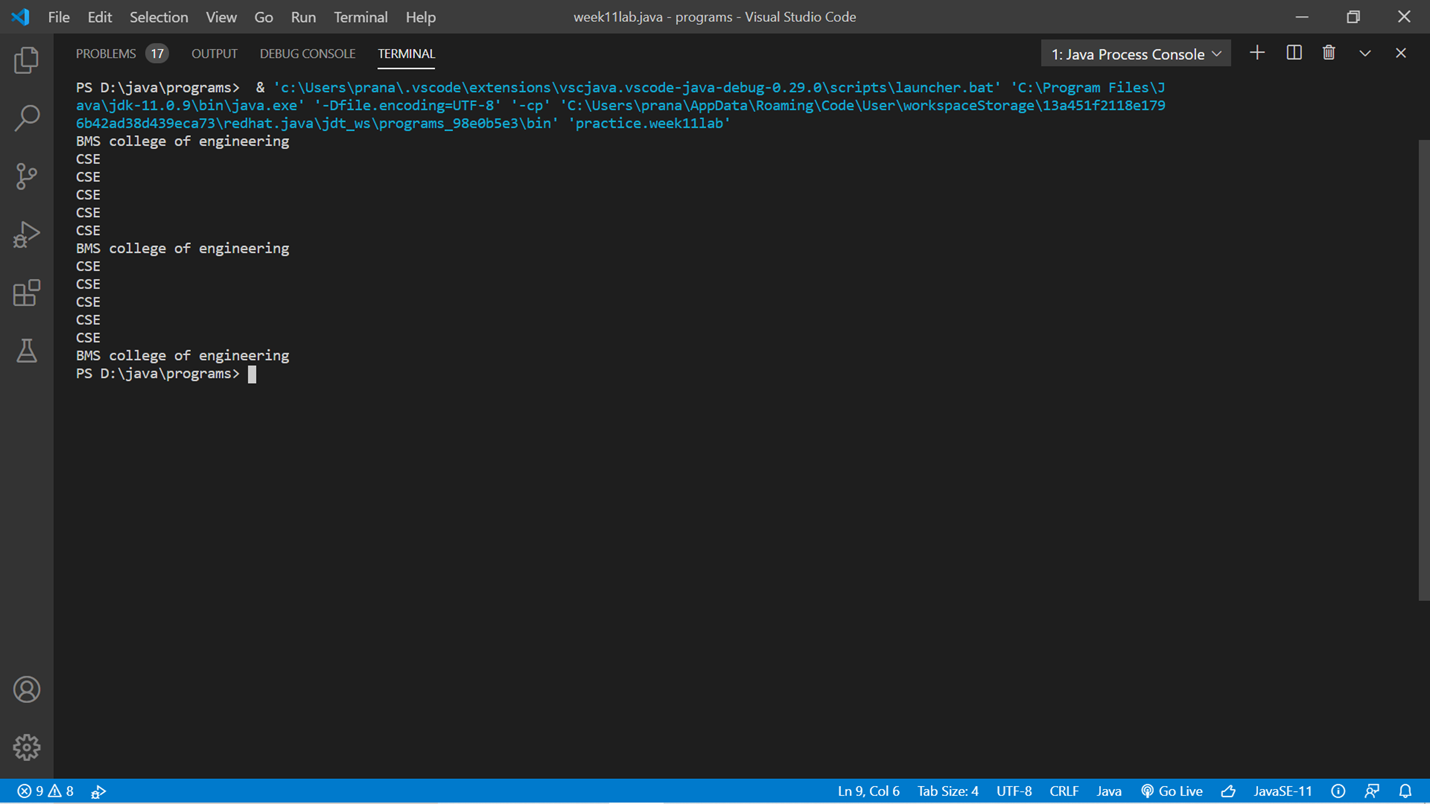
}

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.







package practice;

class BMS implements Runnable{

    Thread T;

    String Name;

    public BMS(String name){

        this.Name=name;

        T=new Thread(this,this.Name);

        T.start();

    }

    public void run(){

        for(int i=0;i<3;i++){

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

            try {

                Thread.sleep(10000);

            } catch (Exception e) {

            }

        }

    }

}

class CSE implements Runnable{

    Thread t;

    String name;

    CSE(String name){

        this.name=name;

        t=new Thread(this,this.name);

        t.start();

    }

    @Override

    public void run() {

        for(int i=0;i<10;i++){

            System.out.println("CSE");

            try {

                Thread.sleep(2000);

            } catch (Exception e) {

            }

        }

    }

}

public class week11lab {

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

        new BMS("Thread 1");

        new CSE("Thread 2");

        // t1.join();

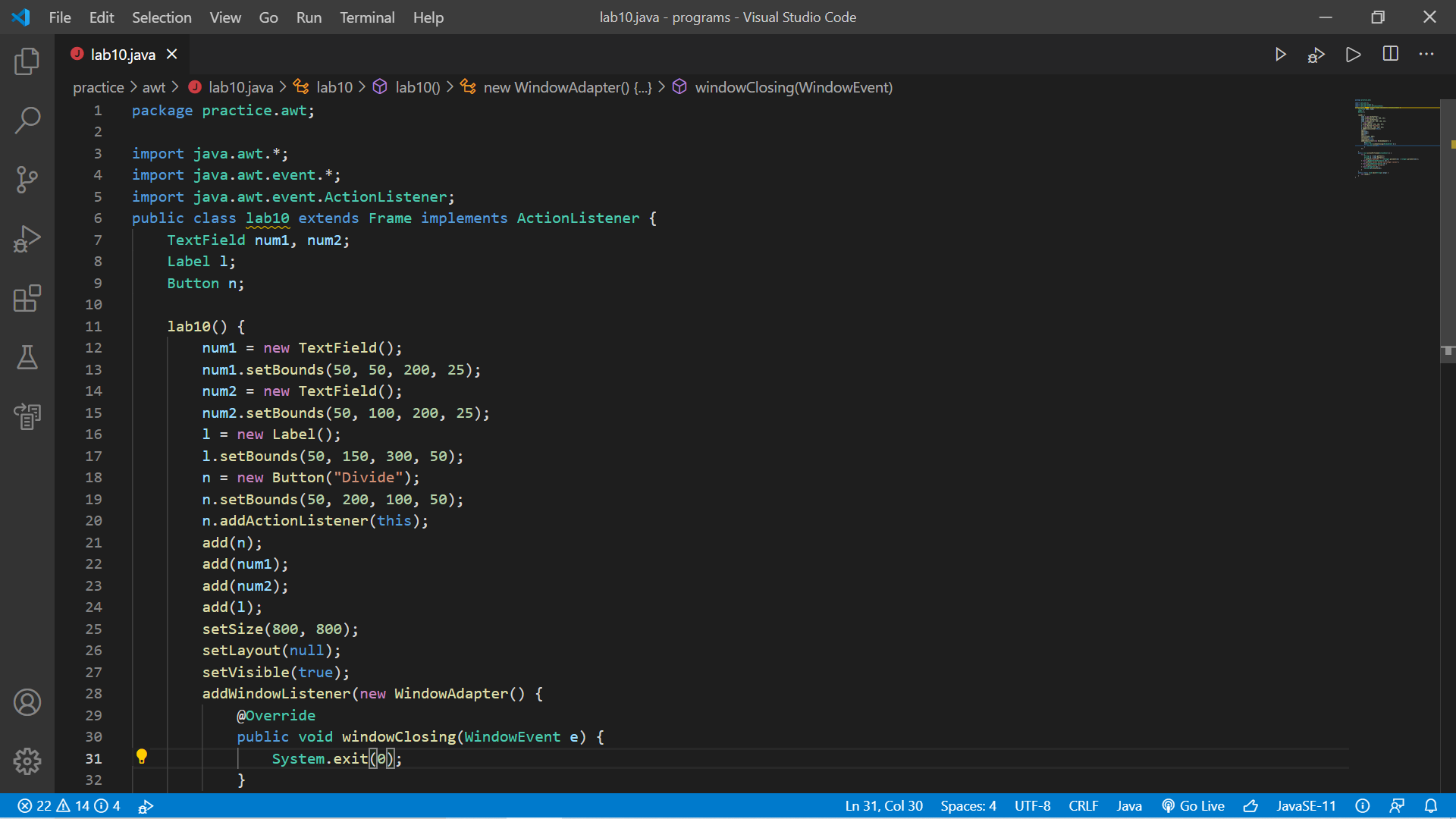
        // t2.join();

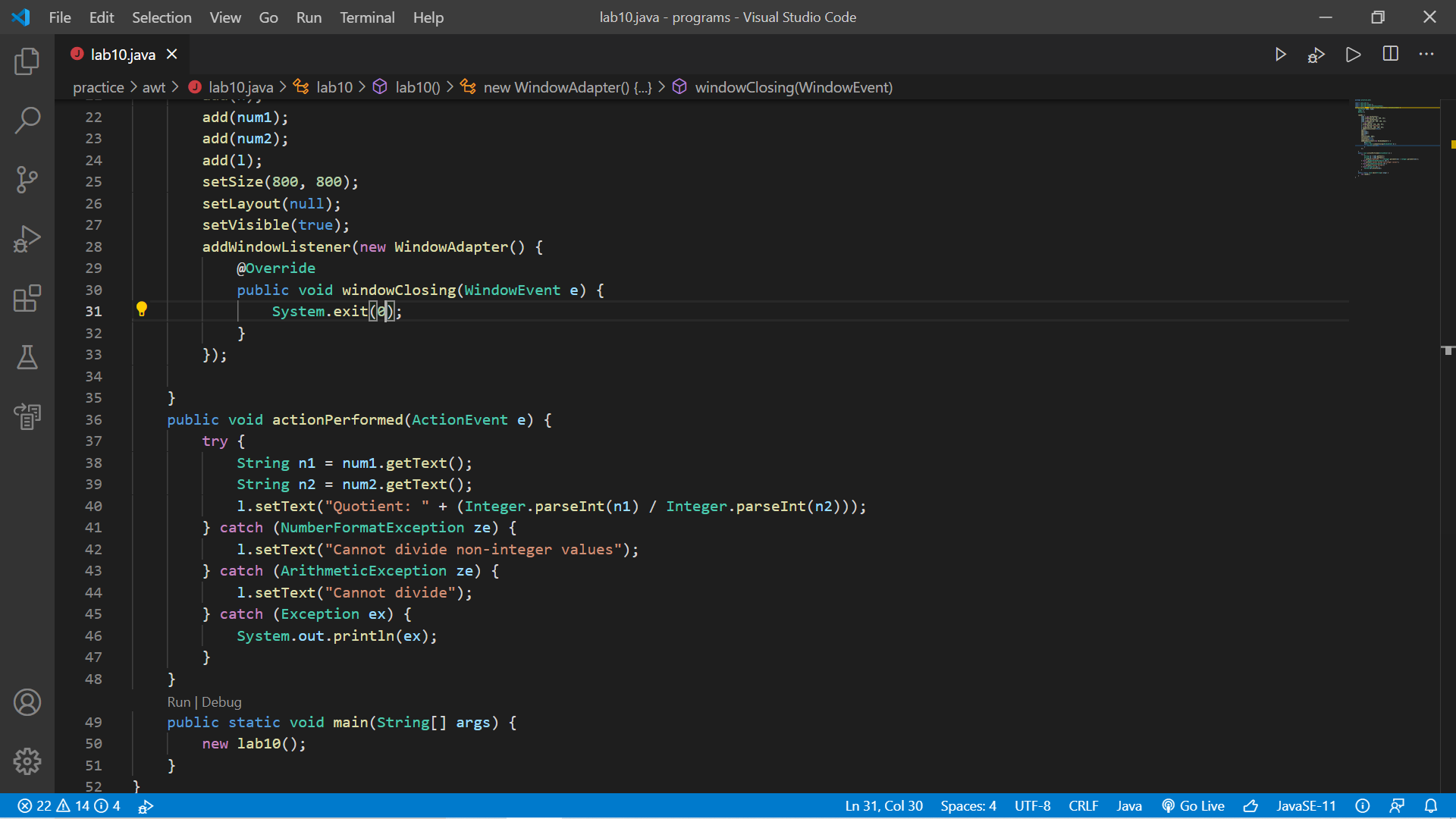
    }

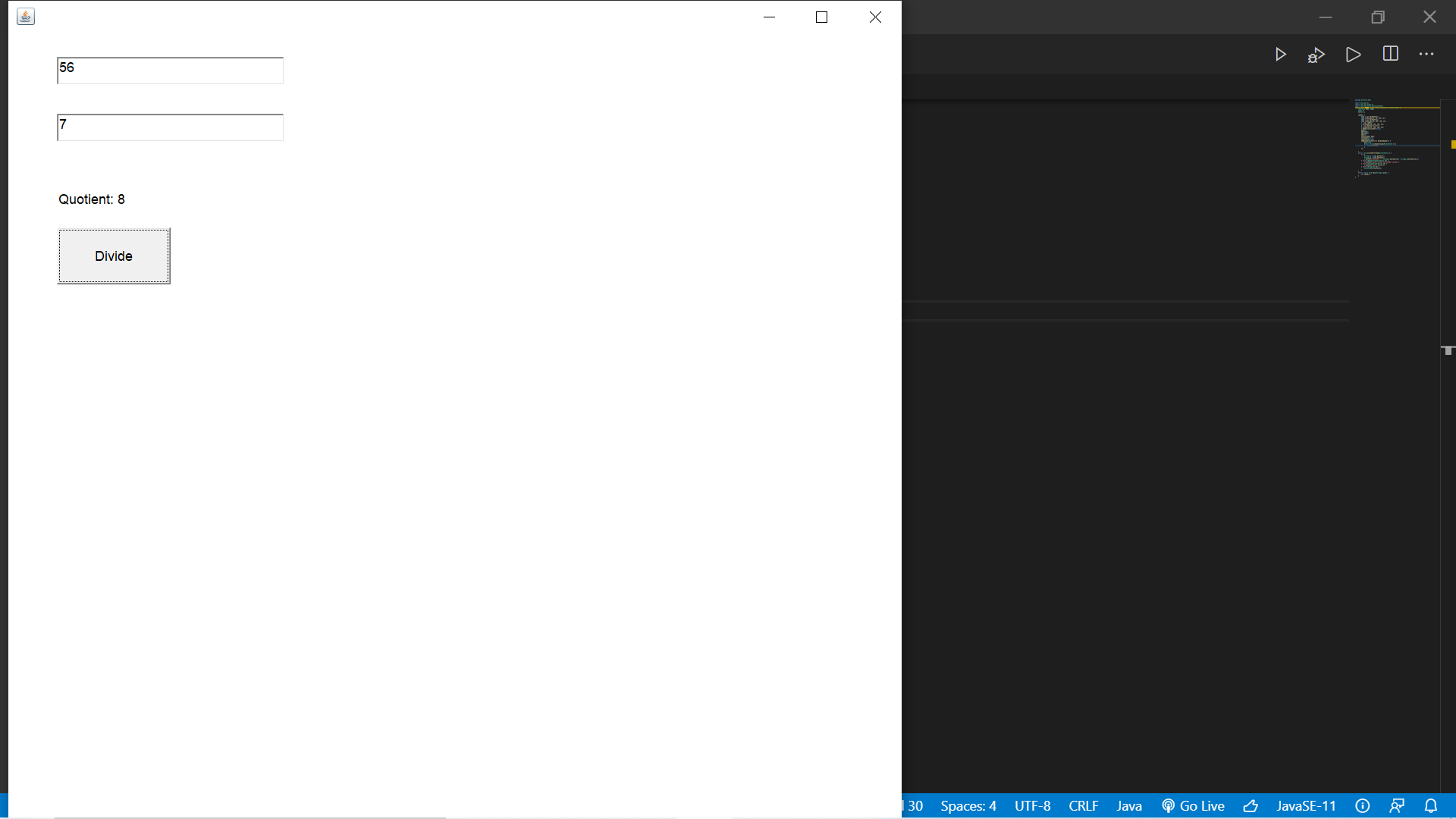
}

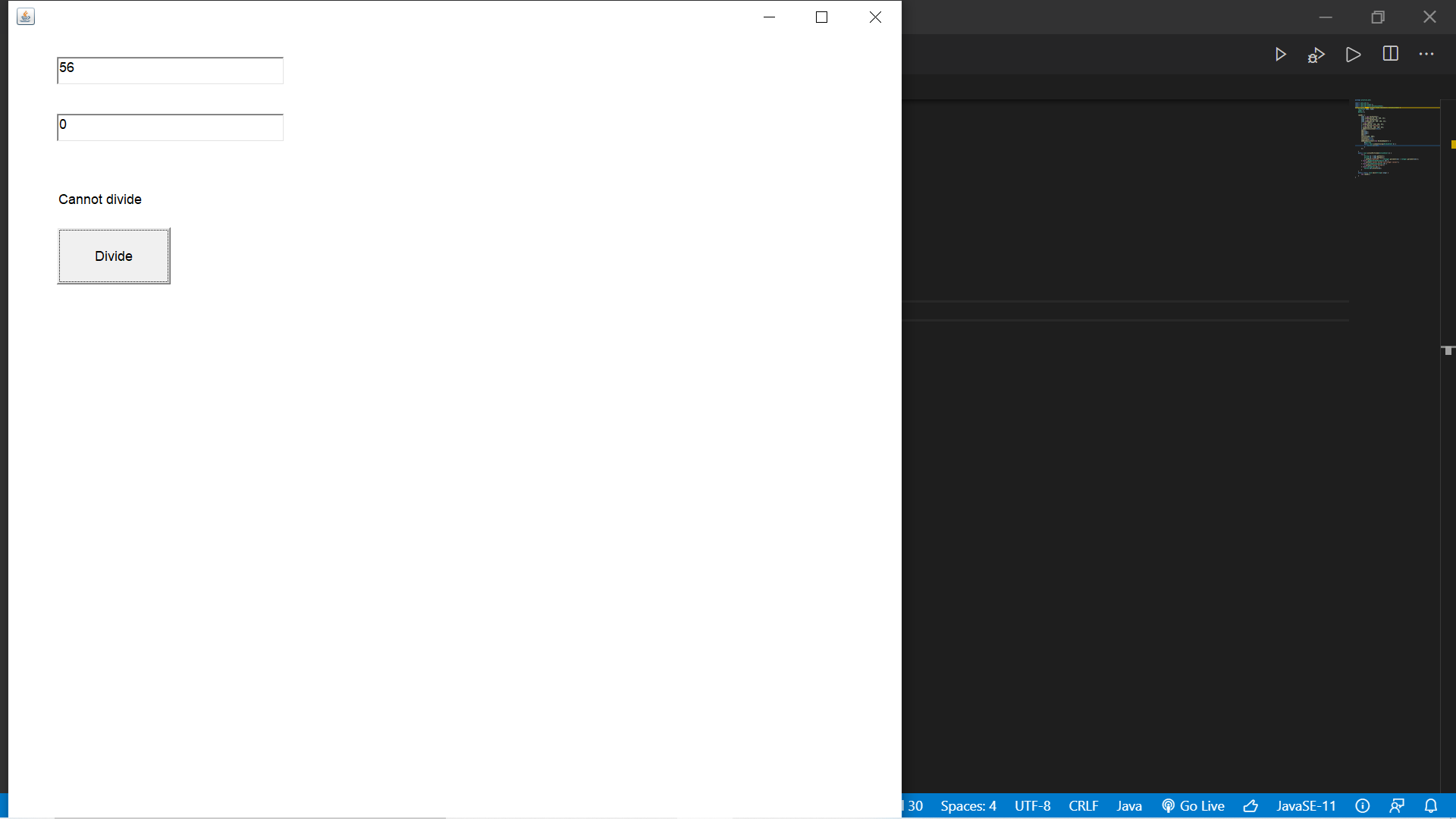
Program 10

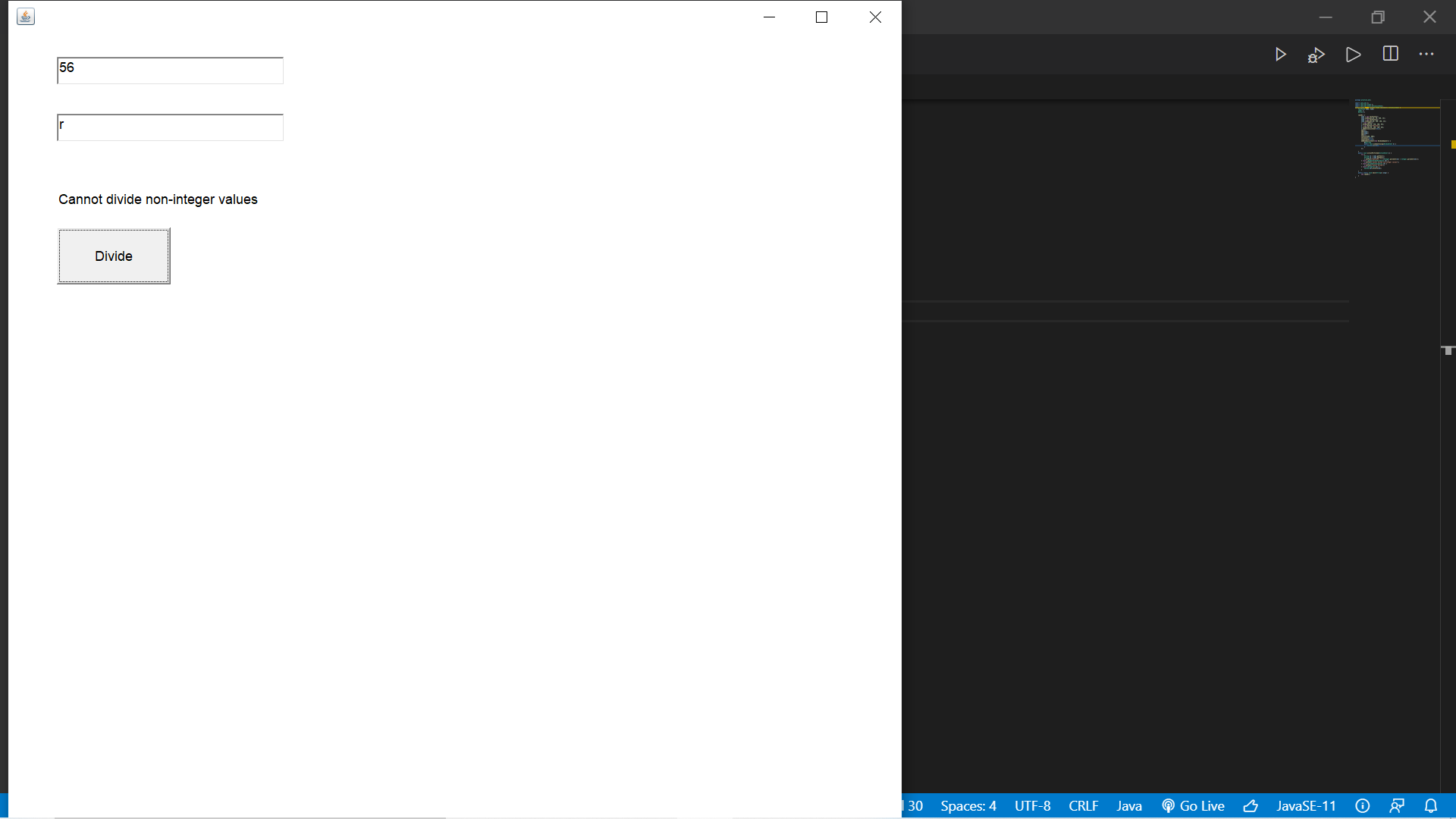
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException.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











package practice.awt;

import java.awt.\*;

import java.awt.event.\*;

import java.awt.event.ActionListener;

public class lab10 extends Frame implements ActionListener {

    TextField num1, num2;

    Label l;

    Button n;

    lab10() {

        num1 = new TextField();

        num1.setBounds(50, 50, 200, 25);

        num2 = new TextField();

        num2.setBounds(50, 100, 200, 25);

        l = new Label();

        l.setBounds(50, 150, 300, 50);

        n = new Button("Divide");

        n.setBounds(50, 200, 100, 50);

        n.addActionListener(this);

        add(n);

        add(num1);

        add(num2);

        add(l);

        setSize(800, 800);

        setLayout(null);

        setVisible(true);

        addWindowListener(new WindowAdapter() {

            @Override

            public void windowClosing(WindowEvent e) {

                System.exit(0);

            }

        });

    }

    public void actionPerformed(ActionEvent e) {

        try {

            String n1 = num1.getText();

            String n2 = num2.getText();

            l.setText("Quotient: " + (Integer.parseInt(n1) / Integer.parseInt(n2)));

        } catch (NumberFormatException ze) {

            l.setText("Cannot divide non-integer values");

        } catch (ArithmeticException ze) {

            l.setText("Cannot divide");

        } catch (Exception ex) {

            System.out.println(ex);

        }

    }

    public static void main(String[] args) {

        new lab10();

    }}