RAVI SAJJANAR
1BM19CS127
CC SECTION BATCH-1
OOJ LAB RECORD SUBMISSION

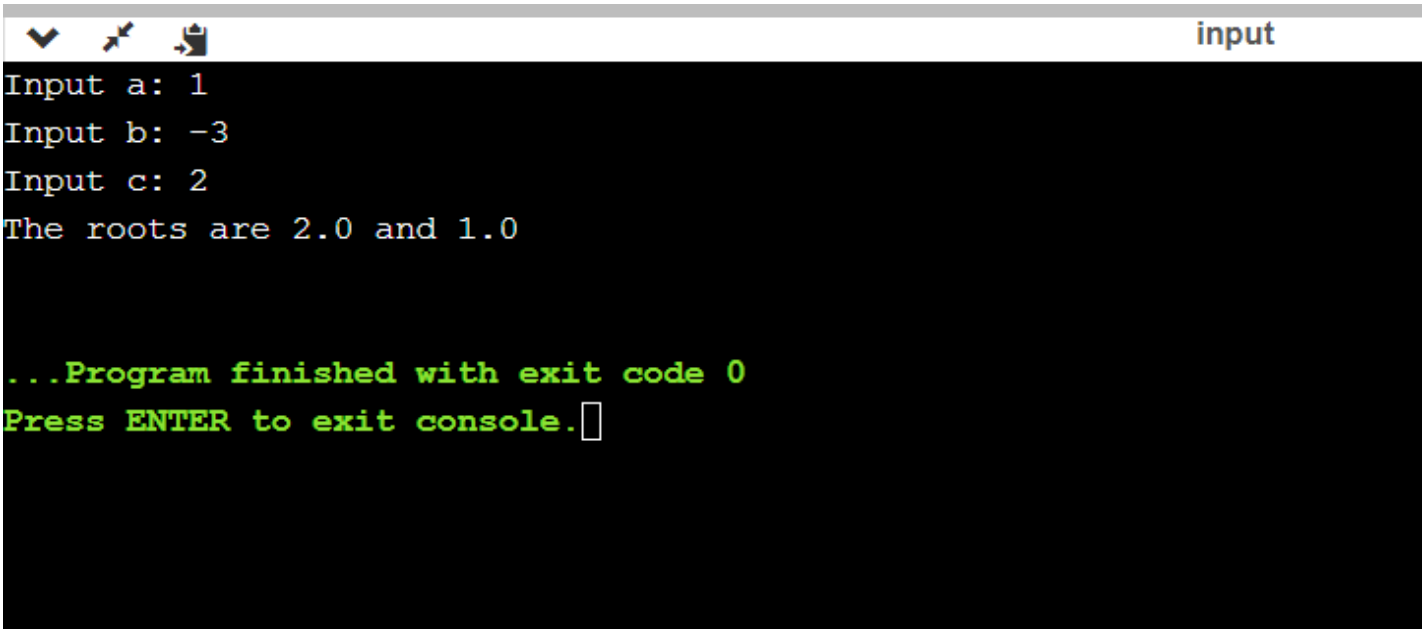
Lab Program: 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;

public class Main {

  public static void main(String[] Strings) {

      Scanner input = new Scanner(System.in);

         System.out.print("Input a: ");

         double a = input.nextDouble();

         System.out.print("Input b: ");

         double b = input.nextDouble();

         System.out.print("Input c: ");

         double c = input.nextDouble();

         double result = b * b - 4.0 * a * c;


         if (result > 0.0) {

            double r1 = (-b + Math.pow(result, 0.5)) / (2.0 * a);

            double r2 = (-b - Math.pow(result, 0.5)) / (2.0 * a);

            System.out.println("The roots are " + r1 + " and " + r2);

         } else if (result == 0.0) {
```

```
            double r1 = -b / (2.0 * a);
            System.out.println("The root is " + r1);
        } else {
            System.out.println("The equation has no real roots.");
        }
    }
}
```

OUTPUT:



Lab Program 2:

Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;

class Student

{

 private String USN;

 private String name;

 private int n;

 private double SGPA = 0;

 private int totalCredits = 0;

 Scanner ss = new Scanner(System.in);


 void Details()

 {

 System.out.println("Enter USN of the student");

 USN = ss.nextLine();

 System.out.println("Enter Name of the student");

 name = ss.nextLine();

 System.out.println("Enter no of subjects");

 n = ss.nextInt();

 int credits[] = new int[n];

 double marks[] = new double[n];

 System.out.println("Enter details of the subjects:");

 for(int i=0;i<n;i++)

 {

  System.out.println("Enter credits allotted to the subject "+(i+1));
```

```
 credits[i] = ss.nextInt();
 System.out.println("Enter marks in the subject "+(i+1));
 marks[i] = ss.nextInt();
 Calculate(credits[i],marks[i],i);
}
  }
  void Calculate(int credit,double mark,int j)
  {
 totalCredits = totalCredits + credit;
 if(mark>=90&&mark<=100)
  SGPA = SGPA + (10*credit);
 else if(mark>=80 && mark<=89)
  SGPA = SGPA + (9*credit);
 else if(mark>=70&&mark<=79)
  SGPA = SGPA + (8*credit);
 else if(mark>=60&&mark<=69)
  SGPA = SGPA + (7*credit);
 else if(mark>=50 && mark<=59)
  SGPA = SGPA + (6*credit);
 else if(mark>=40&&mark<=49)
  SGPA = SGPA + (5*credit);
 else
  System.out.println("Failed in ubject "+(j+1));
}
void Display()
{
 System.out.println("Details of the Student");
 System.out.println("Name :"+name);
 System.out.println("USN: "+USN);
```

```
  System.out.println("SGPA of student "+(SGPA/totalCredits));
 }
}

public class Lab2
{
 public static void main(String args[])
 {
  Student s1 = new Student();
  s1.Details();
  s1.Display();
 }
}
```

OUTPUT:

```
input
Enter Name of the student
RAVI SAJJANAR
Enter no of subjects
4
Enter details of the subjects:
Enter credits allotted to the subject 1
5
Enter marks in the subject 1
86
Enter credits allotted to the subject 2
3
Enter marks in the subject 2
56
Enter credits allotted to the subject 3
2
Enter marks in the subject 3
96
Enter credits allotted to the subject 4
4
Enter marks in the subject 4
63
Details of the Student
Name :RAVI SAJJANAR
USN: 1BM19CS127
SGPA of student 7.928571428571429
```

Lab Program - 3

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

PROGRAM:

```
import java.util.Scanner;

class Book

{

   private String book_name;

   private String author;

   private double price;

   private int no_of_pages;

   Scanner s = new Scanner(System.in);


   public Book() {

      this.book_name = "";

      this.author = "";

      this.price = 0;

      this.no_of_pages = 0;

   }


   void getdetails(){

      System.out.print("\nEnter book name:");

      book_name = s.nextLine();

      System.out.print("Enter author's name:");

      author = s.nextLine();

      System.out.print("Enter price of book:");
```

```
        price = s.nextDouble();
        System.out.print("Enter the number of pages of book:");
        no_of_pages = s.nextInt();
    }
  public String toString()
    {
        return ("------------BOOK DETAILS----------\nBook name:"+ book_name
+"\n"+"Author:"+author+"\n"+ "Price:"+price+"\n"+ "Number of Pages:"+no_of_pages+"\n");
    }
}


public class Main {
    public static void main(String args[]) {
      int n,i=0;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of books:");
        n = sc.nextInt();
        sc.nextLine();
        Book [] b = new Book[n];
        for( i=0;i<n;i++) {
            System.out.print("Enter the detals of book "+(i+1));
            b[i] = new Book();
            b[i].getdetails();
        }
        for(i=0;i<n;i++) {
            System.out.println(b[i]);
        }
    }
}
```

OUTPUT:

```
Enter the number of books:2
Enter the detals of book 1
Enter book name:DS
Enter author's name:SHAMSUNDAR
Enter price of book:236
Enter the number of pages of book:1245
Enter the detals of book 2
Enter book name:OOJ
Enter author's name:KALPATARU
Enter price of book:562
Enter the number of pages of book:889
------------BOOK DETAILS---------------
Book name:DS
Author:SHAMSUNDAR
Price:236.0
Number of Pages:1245

------------BOOK DETAILS---------------
Book name:OOJ
Author:KALPATARU
Price:562.0
Number of Pages:889
```

Lab program4

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes containonly the method printArea( ) that prints the area of the given shape.

```
import java.util.Scanner;
abstract class Shape{
   int a;
   int b;
   Shape(int a,int b){
      this.a=a;
      this.b=b;
   }
   Shape(int a){
      this.a=a;
   }

   abstract double printArea();
}

class Rectangle extends Shape{
   Rectangle(int a,int b){
      super(a,b);
   }

   double printArea(){
```

```
        return a*b;
    }
}

class Triangle extends Shape{
    Triangle(int a,int b){
        super(a,b);
    }

    double printArea(){
        return a*b/2;
    }
}

class Circle extends Shape{
    Circle(int a){
        super(a);
    }

    double printArea(){
        return Math.PI*a*a;
    }
}

  /* public class AbstractShape{
```

```
    public static void main(String args[]){
        Rectangle p=new Rectangle(5,6);
        Triangle q=new Triangle(4,8);
        Circle r=new Circle(3);
    */
    public class AbstractShape{
    public static void main(String args[]){
         Scanner input=new Scanner(System.in);
        System.out.println("Enter the Dimensions of Rectangle");
        Rectangle p=new Rectangle(input.nextInt(),input.nextInt());
        System.out.println("Enter the Dimensions of Triangle");
        Triangle q=new Triangle(input.nextInt(),input.nextInt());
        System.out.println("Enter the Dimensions of Circle");
        Circle r=new Circle(input.nextInt());

    Shape figref;
    figref=p;
    System.out.println("Area of Rectangle is: "+p.printArea());
    figref=q;
    System.out.println("Area of Triangle is: "+q.printArea());
    figref=r;
    System.out.println("Area of Circle is: "+r.printArea());

    }
}
```

Result

CPU Time: 0.27 sec(s), Memory: 37136 kilobyte(s) compiled and executed in 0.936 sec(s)

```
Enter the Dimensions of Rectangle
Enter the Dimensions of Triangle
Enter the Dimensions of Circle
Area of Rectangle is: 24.0
Area of Triangle is: 12.0
Area of Circle is: 50.26548245743669
```

Lab program 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: • Accept deposit from customer and update the balance. • Display the balance. • Compute and deposit interest • Permit withdrawal and update the balance • Check for the minimum balance, impose penalty if necessary and update the balance

```
import java.util.Scanner;


class Account{
	String name;
	int accountNo;
	String accountType;
	double balance;
	Account(String name,int accountNo,String accountType,double balance){
		this.name = name;
		this.accountNo = accountNo;
		this.accountType = accountType;
		this.balance = balance;
	}
	void DisplayStatus() {
		System.out.println("**"+this.accountType+"***");
		System.out.println("Name: "+this.name);
```

```
            System.out.println("Account no.:  "+this.accountNo);
            System.out.println("Account Type:  "+this.accountType);
            System.out.println("Balance: "+this.balance);
        }
}

class SavAcct extends Account{
        double depositAmount;
        double Withdrawmount;
        SavAcct(String name,int accountNo,String accountType,double balance){
            super(name,accountNo,accountType,balance);
        }
        static Scanner input = new Scanner(System.in);
        private void checkBalance() {
            if(balance<0) {
                System.out.println("Transaction is not possible. Balance becomes less
than zero");
                balance+=Withdrawmount;
                Withdrawmount=0;
                Withdraw();
            }
        }
        void CalInterest() {
            System.out.println("Interest To Be added");
            System.out.println("Annual rate of interest: 4%");
            System.out.println("Enter the tenure in terms of year");
```

```
            int tenure = input.nextInt();
            balance = balance*Math.pow(1.04, tenure);
        }
        void Deposit() {
            System.out.println("Enter the Deposit amount");
            depositAmount = input.nextDouble();
            balance+=depositAmount;
        }
        void Withdraw() {
            System.out.println("Enter the Withdrawal amount");
            Withdrawmount = input.nextDouble();
            balance-=Withdrawmount;
            checkBalance();
            System.out.println("Withdraw amount = "+Withdrawmount);
        }
}

class CurrAcct extends Account{
        double minBalance = 1000;
        double depositAmount;
        double Withdrawmount;
        static Scanner input = new Scanner(System.in);
        CurrAcct(String name,int accountNo,String accountType,double balance){
            super(name,accountNo,accountType,balance);
        }
        private void checkBalance() {
```

```
            if(balance<minBalance) {
                System.out.println("Transaction is not possible. Balance becomes less than minimum balance.");
                balance+=Withdrawmount;
                System.out.println("Do u still want to do the transaction with added service charges");
                String ans = input.next();
                if(ans.toLowerCase().equals("yes")) {
                    balance-=(Withdrawmount+(0.05*Withdrawmount)+1000);
                    System.out.println("ALERT: Negative balance.\nService Charge added: "+(0.05*Withdrawmount));
                }else {
                    Withdrawmount = 0;
                }
            }
        }
        void Deposit() {
            System.out.println("Enter the Deposit amount");
            depositAmount = input.nextDouble();
            balance+=depositAmount;
        }
        void Withdraw() {
            System.out.println("Enter the Withdrawal amount");
            Withdrawmount = input.nextDouble();
            balance-=Withdrawmount;
            checkBalance();
            System.out.println("withdraw amount = "+Withdrawmount);
```

```
	}
}

public class Bank {

	public static void main(String[] args) {
		Scanner in = new Scanner(System.in);
		System.out.println("Enter the name");
		String name = in.next();
		System.out.println("Enter the account no.");
		int num = in.nextInt();
		int i=0;
		while(i<2) {
		System.out.println("Enter the account type\ncurr-current acc.\nsav-savings
acct.");
		String type = in.next();
		if(type.equals("curr")) {
			double bal = in.nextInt();
			CurrAcct c1 = new CurrAcct(name,num,"Current Account",bal);
			c1.DisplayStatus();
			c1.Deposit();
			c1.DisplayStatus();
			c1.Withdraw();
			c1.DisplayStatus();
		}else if(type.toLowerCase().equals("sav")) {
			double bal = in.nextInt();
```

```
                SavAcct s1 = new SavAcct(name,num,"Savings Account",bal);
                s1.DisplayStatus();
                s1.Deposit();
                s1.DisplayStatus();
                s1.Withdraw();
                s1.DisplayStatus();
                s1.CalInterest();
                s1.DisplayStatus();
            }
            i++;
            }
            in.close();
        }
}
```

Result

compiled and executed in 120.568 sec(s)

```
Enter the name
Ravi
Enter the account no.
123456789
Enter the account type
curr-current acc.
sav-savings acct.
curr
12302
**Current Account***

Name: RaviAccount no.:  123456789
Account Type:  Current Account
Balance: 12302.0
Enter the Deposit amount
12345
**Current Account***
Name: Ravi
Account no.:  123456789
Account Type:  Current Account
Balance: 135758.0
Enter the Withdrawal amount
456892
Transaction is not possible. Balance becomes less than minimum balance.
Do u still want to do the transaction with added service charges
yes
ALERT: Negative balance.
Service Charge added: 22844.600000000002
withdraw amount = 456892.0
**Current Account***
Name: Ravi
Account no.:  123456789
Account Type:  Current Account
Balance: -344978.6
Enter the account type
curr-current acc.
sav-savings acct.
```

Lab program 6

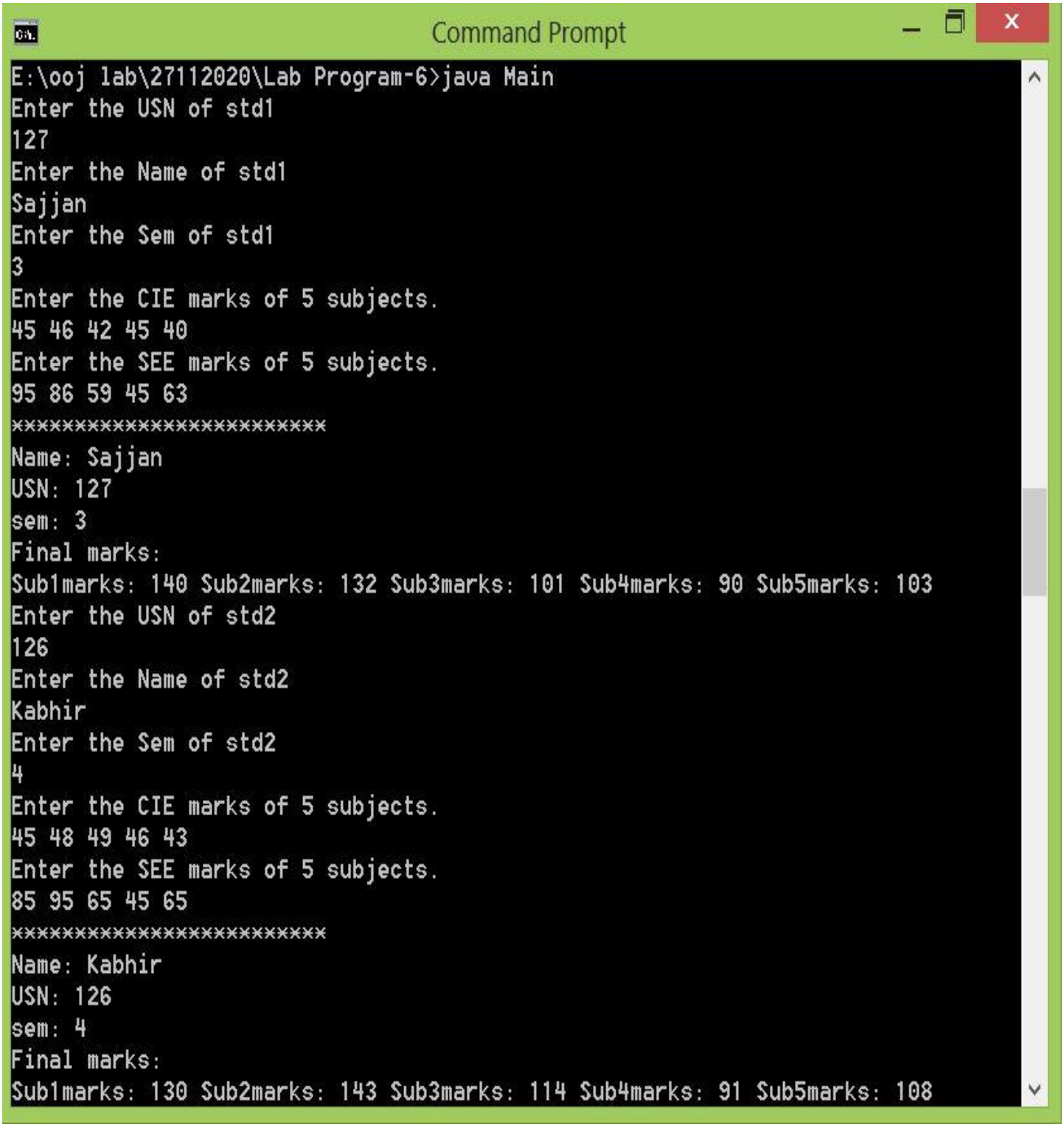
Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External 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.

```
// Internals class
//package cie

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;

}

}

// Student Class
//package cie


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 see
//class externals

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 Main{

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++){

System.out.println("Enter the USN of std"+(i+1));
int usn1 = input.nextInt();

System.out.println("Enter the Name of std"+(i+1));
String name1 = input.next();

System.out.println("Enter the Sem of std"+(i+1));
int sem1 = input.nextInt();

System.out.println("Enter the CIE marks of 5 subjects.");
int[][] cie = new int[2][5];

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

for(int j=0;j<5;j++){

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

}

System.out.println("Enter the SEE marks of 5 subjects.");
for(int j=0;j<5;j++){

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

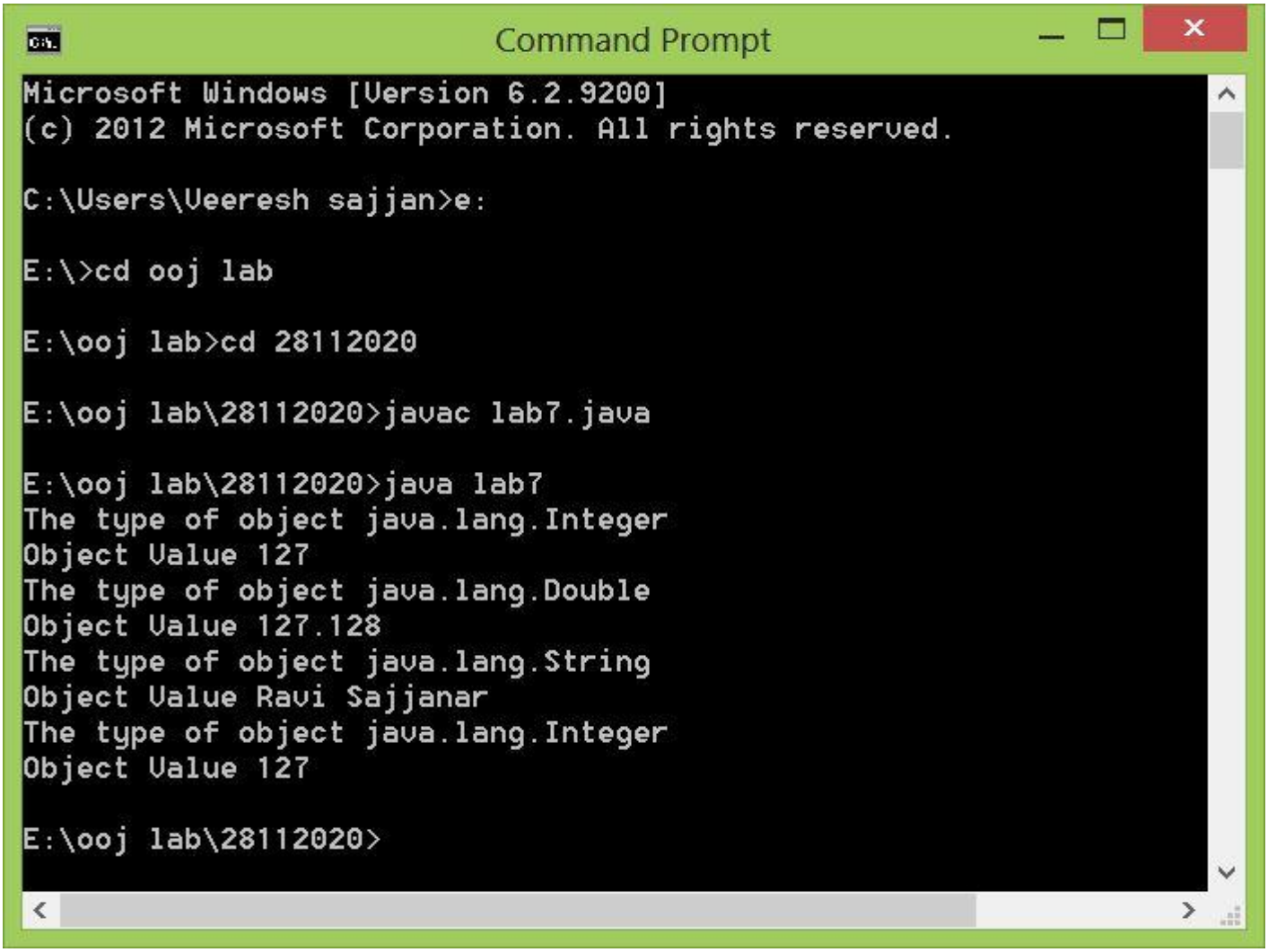
```
}

System.out.println("**************************");
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);

int k=1;
System.out.println("Final marks:");

for(int j=0;j<5;j++){

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

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

}

System.out.println();

 }

}

}
```


Command Prompt
E:\ooj lab\27112020\Lab Program-6>java Main
Enter the USN of std1
127
Enter the Name of std1
Sajjan
Enter the Sem of std1
3
Enter the CIE marks of 5 subjects.
45 46 42 45 40
Enter the SEE marks of 5 subjects.
95 86 59 45 63
****
Name: Sajjan
USN: 127
sem: 3
Final marks:
Sub1marks: 140 Sub2marks: 132 Sub3marks: 101 Sub4marks: 90 Sub5marks: 103
Enter the USN of std2
126
Enter the Name of std2
Kabhir
Enter the Sem of std2
4
Enter the CIE marks of 5 subjects.
45 48 49 46 43
Enter the SEE marks of 5 subjects.
85 95 65 45 65
****
Name: Kabhir
USN: 126
sem: 4
Final marks:
Sub1marks: 130 Sub2marks: 143 Sub3marks: 114 Sub4marks: 91 Sub5marks: 108

Lab program 7

Write a program to demonstrate generics with multiple object parameters.

```
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>(127,127.128);
        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>("Ravi Sajjanar",127);
        ob2.objType();
        System.out.println("Object Value "+ob2.getObj());
        ob2.objType1();
        System.out.println("Object Value "+ob2.getObj1());
    }

}
```

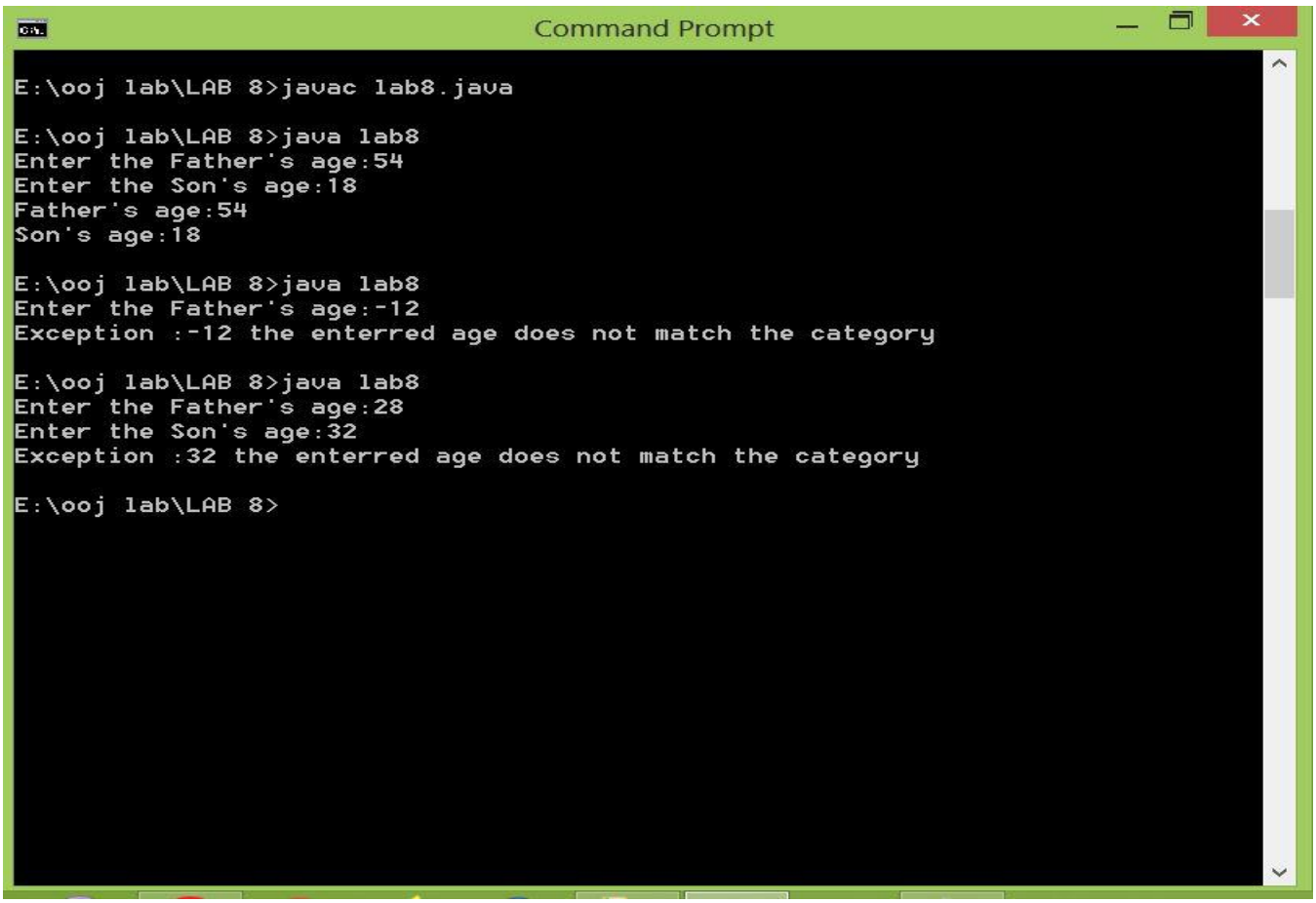

Command Prompt
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.
C:\Users\Veeresh sajjan>e:
E:\>cd ooj lab
E:\ooj lab>cd 28112020
E:\ooj lab\28112020>javac lab7.java
E:\ooj lab\28112020>java lab7
The type of object java.lang.Integer
Object Value 127
The type of object java.lang.Double
Object Value 127.128
The type of object java.lang.String
Object Value Ravi Sajjanar
The type of object java.lang.Integer
Object Value 127
E:\ooj lab\28112020>

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 WrongAge( ) 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.

```
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;
        if(this.age<=0){
            throw new ageException(this.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;
        if(this.age>=this.f.age){
            throw new ageException(this.age);
```

```
        }
    }
    void display(){
        this.f.display();
        System.out.println("Son's age:"+this.age);
    }
}

public class lab8{
    public static void main(String[] args){
        try{
            Scanner input = new Scanner(System.in);
            System.out.print("Enter the Father's age:");
            Father f = new Father(input.nextInt());
            System.out.print("Enter the Son's age:");
            Son s = new Son(input.nextInt(),f);
            s.display();
        }catch(Exception e){
            System.out.println(e);
        }
    }
}
```

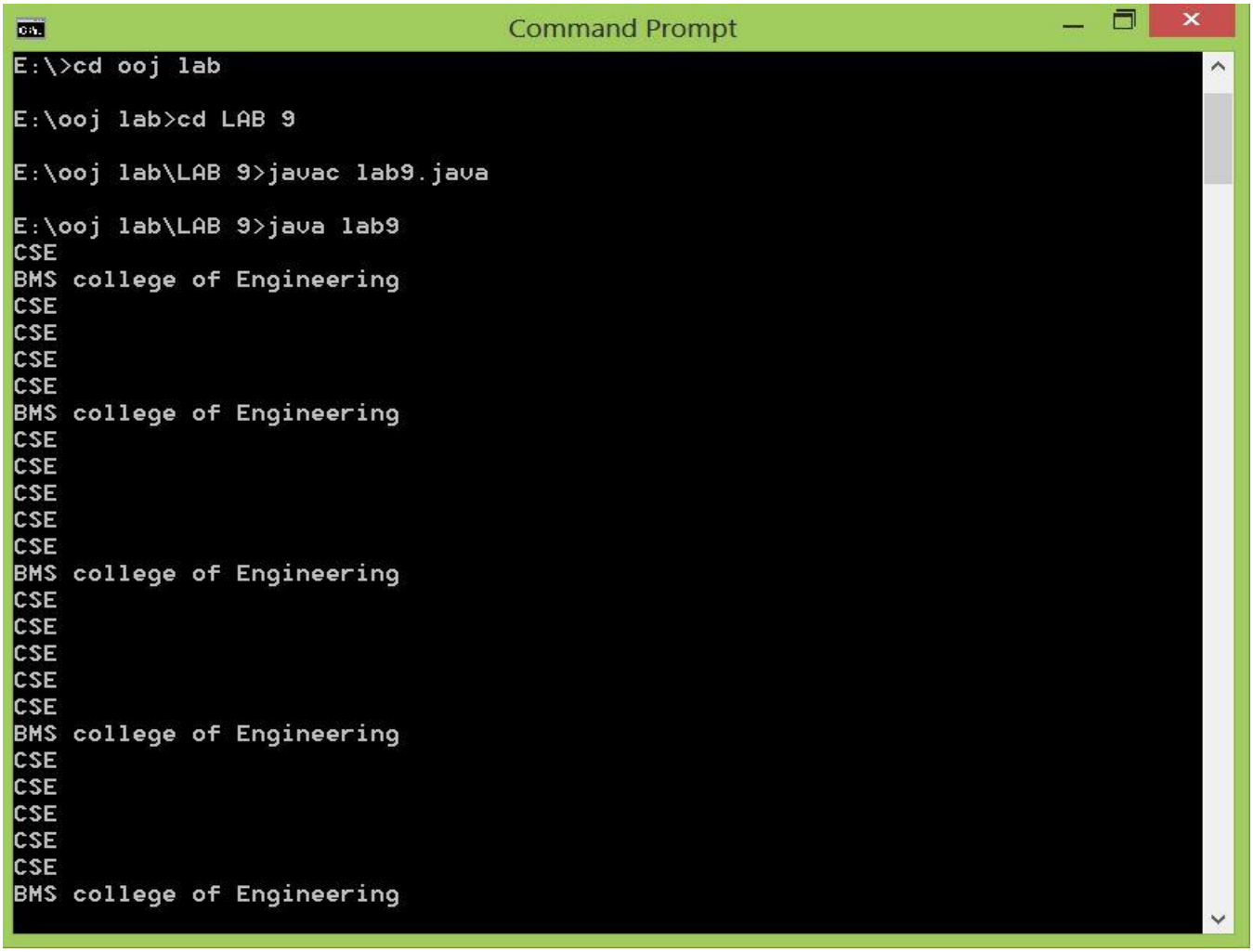


Lab program 9

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.

```
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");
                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 lab9{
    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");
        }
    }
}
```



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.

```
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);
	}
}
```