B.M.S College of Engineering

(Autonomous Institution affiliated to VTU, Belagavi)
Bengaluru – 20

Department of Computer Science and Engineering

Report on

OOJP LAB PROGRAMS

Course Title:
Object Oriented Java Programming

Course Code: 19CS3PCOOJ

(Autonomous Scheme 2020)

Submitted by

Name: SUDESHNA BHUSHAN
USN: 1BM19CS189

LAB PROGRAM 1

QUESTION

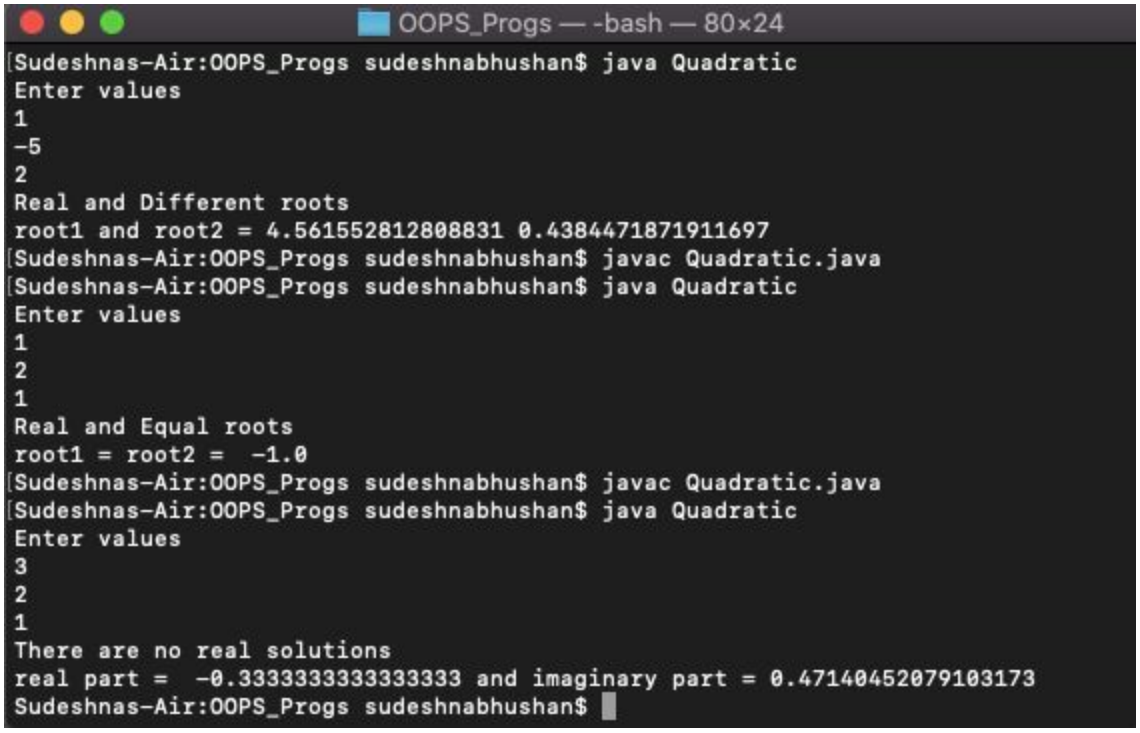
Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

SOLUTION

```
import java.util.*;
public class Quadratic {

    public static void main(String[] args)
    {

        double a, b, c;
        double root1, root2;
        System.out.println("Enter values");
        Scanner input = new Scanner (System.in);
        a = input.nextDouble();
        b = input.nextDouble();
        c = input.nextDouble();

        double determinant = b * b - 4 * a * c;

        // condition for real and different roots
        if(determinant > 0)
```

```
        {
            root1 = (-b + Math.sqrt(determinant)) / (2 * a);
            root2 = (-b - Math.sqrt(determinant)) / (2 * a);
            System.out.println("Real and Different roots");

            System.out.println("root1 and root2 ="+" " + root1 + " " + root2);
        }
        // Condition for real and equal roots
        else if(determinant == 0)
        {
            root1 = root2 = -b / (2 * a);
            System.out.println("Real and Equal roots");


            System.out.println("root1 = root2 = " +" "+ root1);
        }
        // If roots are not real
        else
        {
            double realPart = -b / (2 *a);
            double imaginaryPart = Math.sqrt(-determinant) / (2 * a);

            System.out.println("There are no real solutions");

            System.out.println("real part = "+" "+realPart+" "+"and imaginary part ="+"
"+imaginaryPart);
        }
    }
}
```

OUTPUT



LAB PROGRAM 2

QUESTION

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.

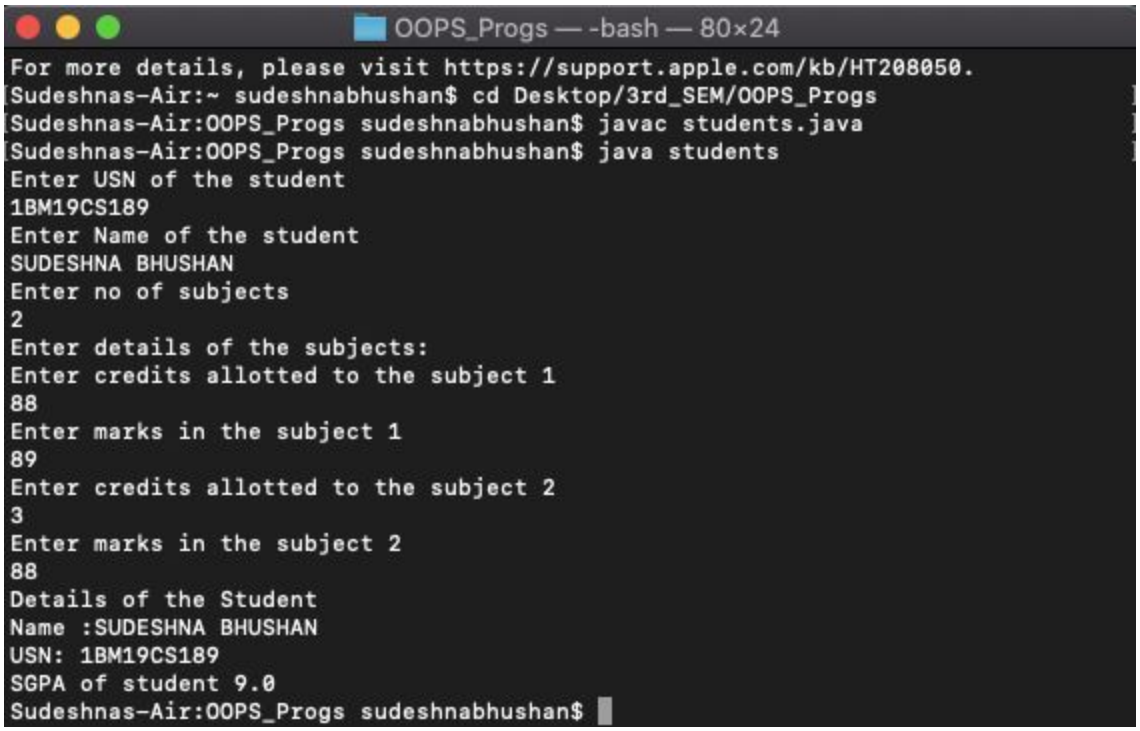
SOLUTION

```
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 subject "+(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 students
{
 public static void main(String args[])
 {
  Student s1 = new Student();
  s1.Details();
  s1.Display();
 }
}
```

OUTPUT



LAB PROGRAM 3

QUESTION

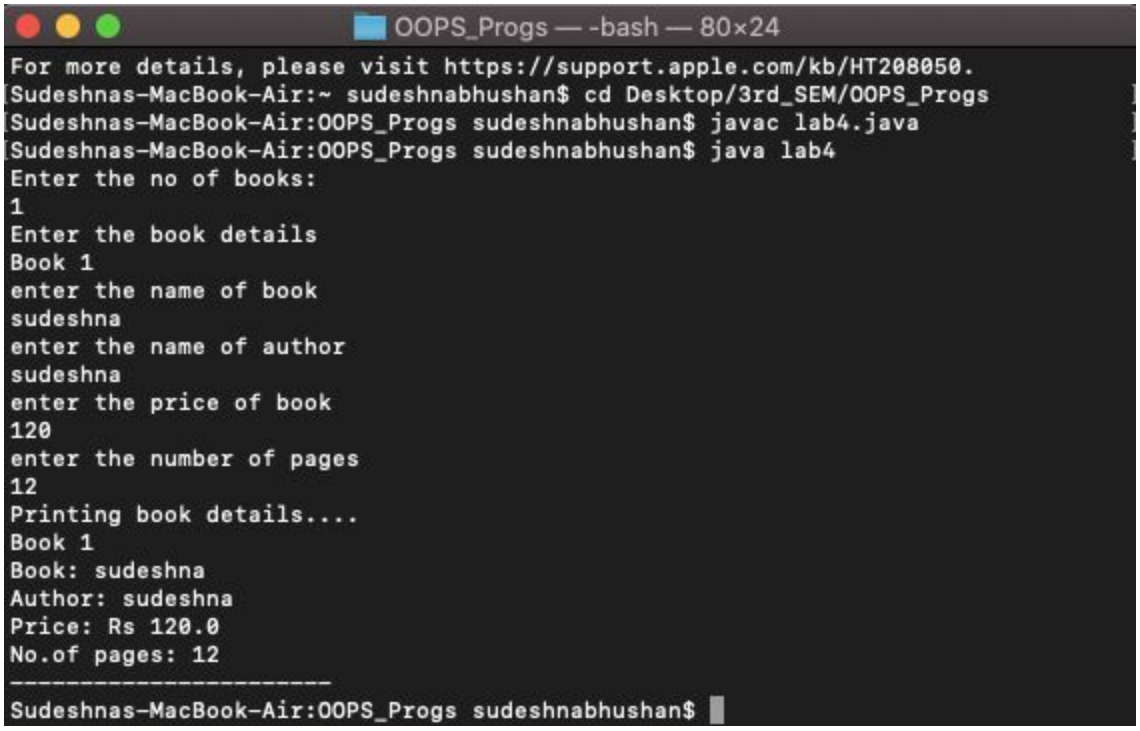
Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program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

SOLUTION

```
import java.util.*;
class Book {
	String name;
	String author;
	int price;
	int num_pages;
	Book()
	{}
	Book(String name,String author,int price,int num_pages)
	{
		this.name=name;
		this.author=author;
		this.price=price;
		this.num_pages=num_pages;
	}
	void accept()
	{
		Scanner s=new Scanner(System.in);
		System.out.println("Enter the name of the book");
		name=s.next();
		System.out.println("Enter the author of the book");
		author=s.next();
		System.out.println("Enter the price of the book");
		price=s.nextInt();
		System.out.println("Enter the number of pages of the book");
		num_pages=s.nextInt();
	}
	public String toString()
	{
		return ("Name: "+name + "\n" + "Author: "+author + "\n" + "Price: "+price + "\n"
+"Number of pages: "+num_pages );
	}
}
class BookMain {
```

```
	public static void main(String ss[])
	{
		Scanner a=new Scanner(System.in);
		Book b1=new Book("Heights","Anne",299,345);
		System.out.println("Sample input:\n"+b1);
		System.out.println("Enter the number of books");
		int n=a.nextInt();
		Book b[]=new Book[n];
		for(int i=0;i<n;i++)
		{
			b[i]=new Book();
			System.out.println("Enter the details of "+(i+1)+" book");
			b[i].accept();
		}
		for(int i=0;i<n;i++)
		{
			System.out.println("Details of book "+(i+1));
			System.out.println(b[i]);
		}
	}
}
```

OUTPUT



LAB PROGRAM 4

QUESTION

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 contain only the method printArea( ) that prints the area of the given shape.

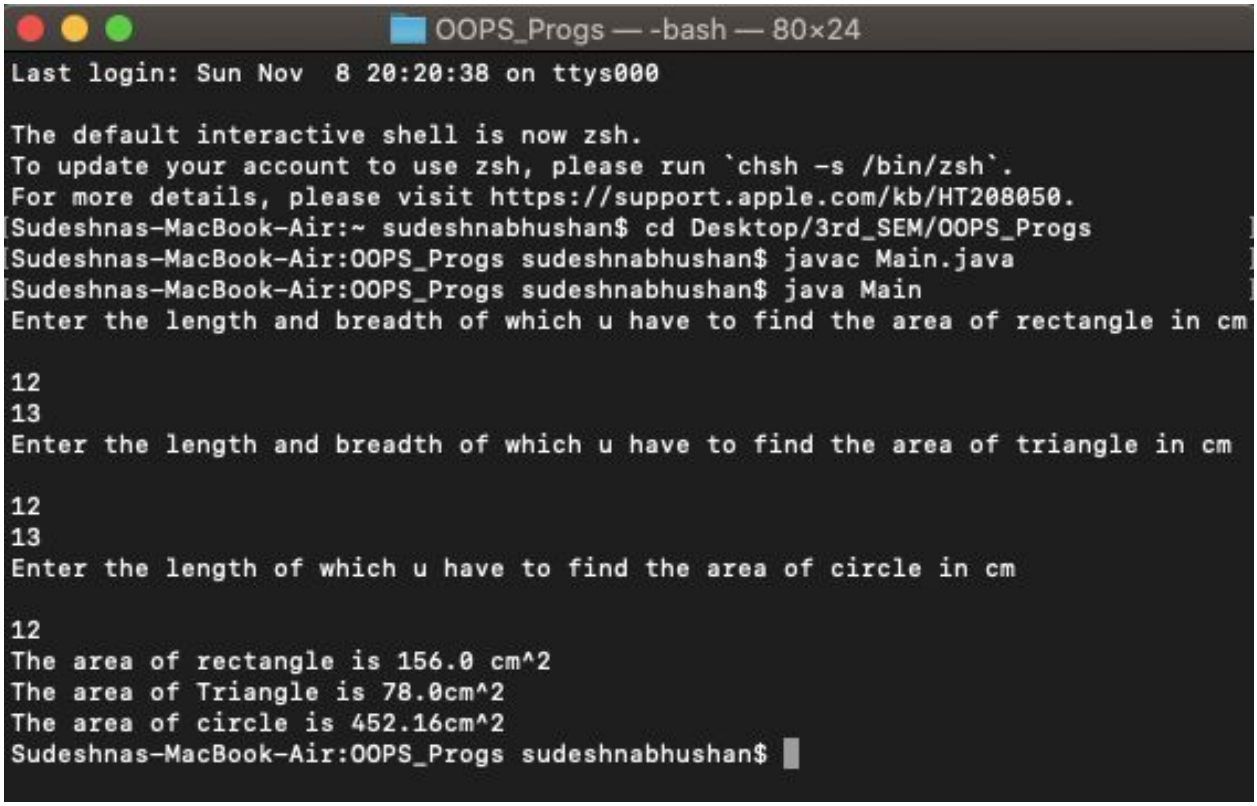
SOLUTION

```
import java.util.Scanner;
```

```
abstract class Shape
{
  int length,breadth;
void printArea()
{}
}
class Rectangle extends Shape
{
double areaR;
void printArea(){
areaR=(length*breadth);
System.out.println("The area of rectangle is "+areaR+" cm^2");
}
}
class Triangle extends Shape
{

double areaT;
void printArea(){
areaT=(0.5)*(length*breadth);
System.out.println("The area of Triangle is "+areaT+"cm^2");
}
}
class Circle extends Shape
{

double areaC;
void printArea(){
areaC=(3.14)*(length*length);
System.out.println("The area of circle is "+areaC+"cm^2");
}
}
class Main
{
public static void main(String args[])
{
Scanner A=new Scanner(System.in);
Rectangle R1=new Rectangle();
Triangle T1=new Triangle();
Circle C1=new Circle();
System.out.println("Enter the length and breadth of which u have to find the area of rectangle in
cm\n");
R1.length=A.nextInt();
R1.breadth=A.nextInt();
```

```
System.out.println("Enter the length and breadth of which u have to find the area of triangle in cm\n");
T1.length=A.nextInt();
T1.breadth=A.nextInt();
System.out.println("Enter the length of which u have to find the area of circle in cm\n");
C1.length=A.nextInt();
R1.printArea();
T1.printArea();
C1.printArea();
}
}
```

OUTPUT



LAB PROGRAM 5

QUESTION

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.

SOLUTION

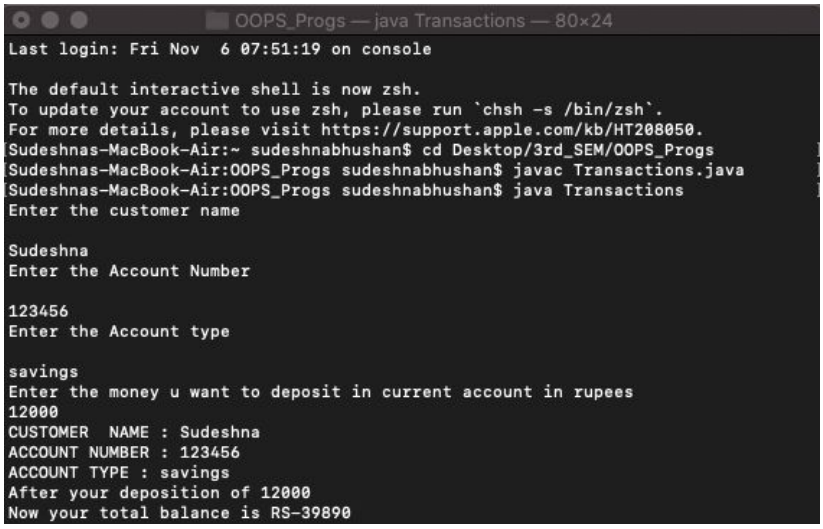
```
import java.util.Scanner;
class Bank
{
int deposit_balance;
int wthdraw_balance;
String customername;
String Account_Number;
String Account_Type;
int Balance=27890;
void accept()
{
Scanner s=new Scanner(System.in);
System.out.println("Enter the customer name\n");
customername=s.next();
System.out.println("Enter the Account Number\n");
Account_Number=s.next();
System.out.println("Enter the Account type\n");
Account_Type=s.next();
}
void display()
{
System.out.println("CUSTOMER  NAME : "+customername);
```

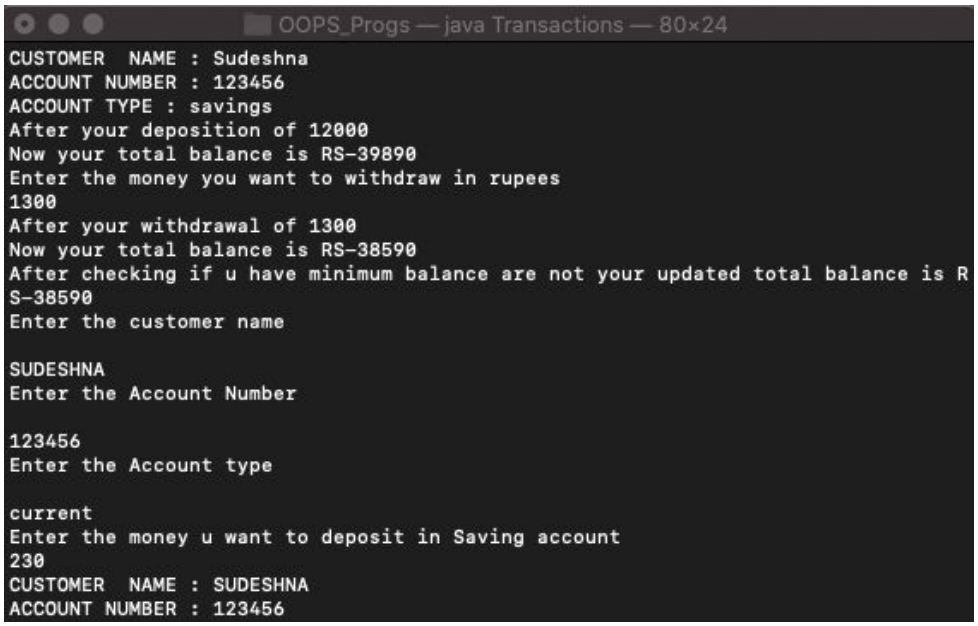
```
System.out.println("ACCOUNT NUMBER : "+Account_Number);
System.out.println("ACCOUNT TYPE : "+Account_Type);

}

}
class curr_acct extends  Bank{

int updated_balance;
int After_cwithdrawn;
int   updated_lost_cbalance;


int cdepo_ba(){

updated_balance=Balance+deposit_balance;
return updated_balance;
}
int cwith_ba(){
After_cwithdrawn=((updated_balance)-(wthdraw_balance));
return After_cwithdrawn;
}
int minimum()
{
if((After_cwithdrawn)<=(2000))
{
updated_lost_cbalance=((After_cwithdrawn)-(200));
System.out.println("you have minimum balance below 2000 so u have lost 200 rs");

return updated_lost_cbalance;

}

else
return After_cwithdrawn;


}

}
class sav_acct extends Bank{

int supdated_balance;
int  After_swithdrawn;
```

```
int updated_lost_sbalance;
int compound_interest;
int sdepo_ba(){

supdated_balance=Balance+deposit_balance;
return supdated_balance;
}
int interest()
{
double r=0.08;
int n=12;
int t=5;
compound_interest=(int)((supdated_balance)*(Math.pow((1+(r/n)),(n*t))));
return compound_interest;
}

int Swith_ba(){
After_swithdrawn=((compound_interest)-(wthdraw_balance));
return After_swithdrawn;
}

int minimum()
{
if((After_swithdrawn)<=(1000))
{
updated_lost_sbalance=((After_swithdrawn)-(100));
return updated_lost_sbalance;

}

else
return After_swithdrawn;
}
}

class Transactions{
public static void main(String args[]){
Scanner r=new Scanner(System.in);
curr_acct CA=new curr_acct();
CA.accept();
System.out.println("Enter the money u want to deposit in current account in rupees");
CA.deposit_balance=r.nextInt();
CA.display();
```

```
System.out.println("After your deposition of "+CA.deposit_balance+"\nNow your total balance is RS-"+CA.cdepo_ba());
System.out.println("Enter the money you want to withdraw in rupees");
CA.wthdraw_balance=r.nextInt();
System.out.println("After your withdrawal of "+CA.wthdraw_balance+"\nNow your total balance is RS-"+CA.cwith_ba());
System.out.println("After checking if u have minimum balance are not your updated total balance is RS-"+CA.minimum());
sav_acct SA=new sav_acct();
SA.accept();
System.out.println("Enter the money u want to deposit in Saving account");
SA.deposit_balance=r.nextInt();
SA.display();
System.out.println("After your deposition of "+SA.deposit_balance+"\nNow your total balance is RS-"+SA.sdepo_ba());
System.out.println("After interest ur updated balance is RS-"+SA.interest());
System.out.println("Enter the money you want to withdraw in Saving account");
SA.wthdraw_balance=r.nextInt();
System.out.println("After your withdrawal of RS-"+SA.wthdraw_balance+"\nNow your total balance is RS-"+SA.Swith_ba());

System.out.println("After checking if u have minimum balance are not your updated total balance is RS-"+SA.minimum());


}
}
```

OUTPUT





LAB PROGRAM 6

QUESTION

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.

SOLUTION

Internals.java

```
package CIE;
import java.util.Scanner;
public class Internals extends CIE.Student
{
    public int ciem[]=new int[5];
    Scanner xx =new Scanner (System.in);
    public void accept()
    {
        for(int i=0;i<5;i++)
        {
            System.out.println("Enter the cie marks of subject"+(i+1)+" out of 50");
            {
                ciem[i]=xx.nextInt();
            }
        }
    }


}
```

Externals.java

```
package SEE;
import CIE.*;
import java.util.Scanner;

public class Externals extends CIE.Student
{
    public int seem[]=new int[5];
    Scanner xx =new Scanner (System.in);
    public void accept()
    {
        for(int i=0;i<5;i++)
        {
            System.out.println("Enter the see marks of subject"+(i+1)+" out of 100");
```

```
        {
            seem[i]=xx.nextInt();
        }
    }
  }

}
```

Student.java

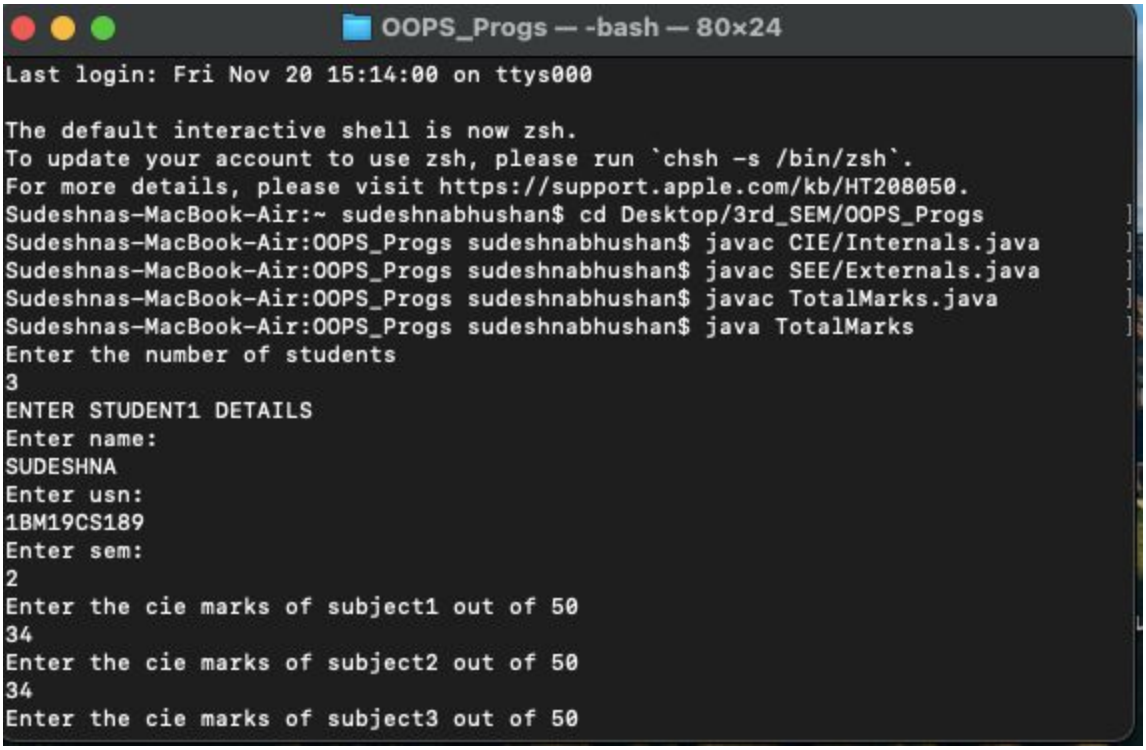
```
package CIE;
import java.util.Scanner;

public class Student
{
   String name,usn;
   int sem;
   Scanner xx=new Scanner(System.in);
   public void accept()
   {
      System.out.println("Enter name:");
      name=xx.nextLine();
      System.out.println("Enter usn:");
      usn=xx.next();
      System.out.println("Enter sem:");
      sem=xx.nextInt();
   }
   public void display()
   {
      System.out.println("Name :"+name);
      System.out.println("Usn :"+usn);
      System.out.println("Sem :"+sem);
   }
}
```

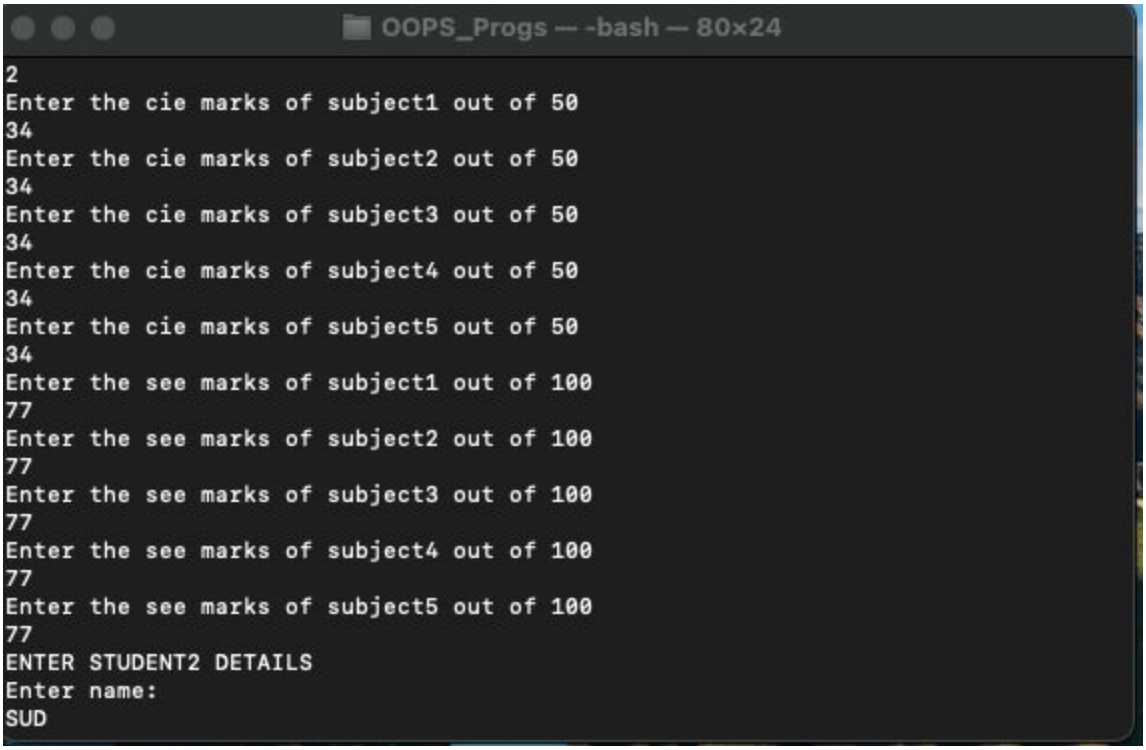
TotalMarks.java

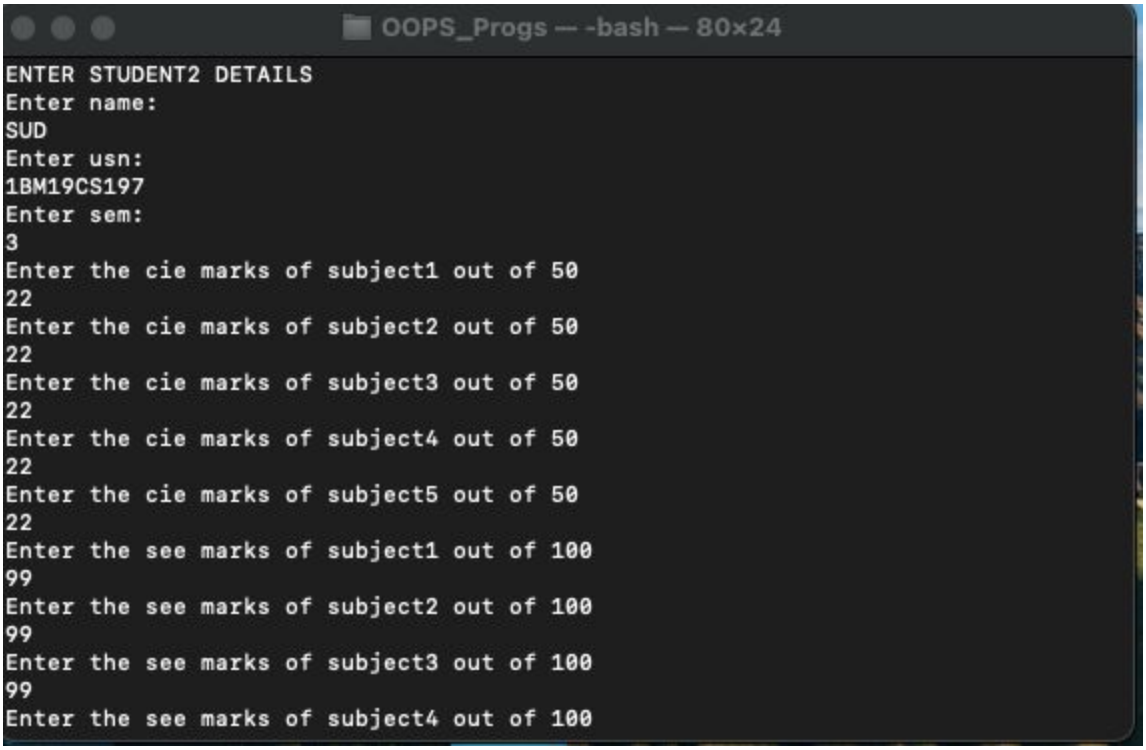
```
import CIE.*;
import SEE.*;
import java.util.*;

class TotalMarks
{
    public static void main(String sss[])
    {
        int i,j,n;
        int total[]=new int[5];
        Scanner xx=new Scanner(System.in);
        System.out.println("Enter the number of students");
        n=xx.nextInt();
        CIE.Student s[]=new CIE.Student[n];
        CIE.Internals ci[]= new CIE.Internals[n];
              SEE.Externals se[]=new SEE.Externals[n];
        for(i=0;i<n;i++)
        {
            System.out.println("ENTER STUDENT"+(i+1)+" DETAILS");
            s[i]=new CIE.Student();
            s[i].accept();
            ci[i]=new CIE.Internals();
            ci[i].accept();
            se[i]=new SEE.Externals();
            se[i].accept();
        }
        for(i=0;i<n;i++)
        {
            System.out.println("DETAILS OF STUDENT "+(i+1));
            s[i].display();
            for(j=0;j<5;j++)
```

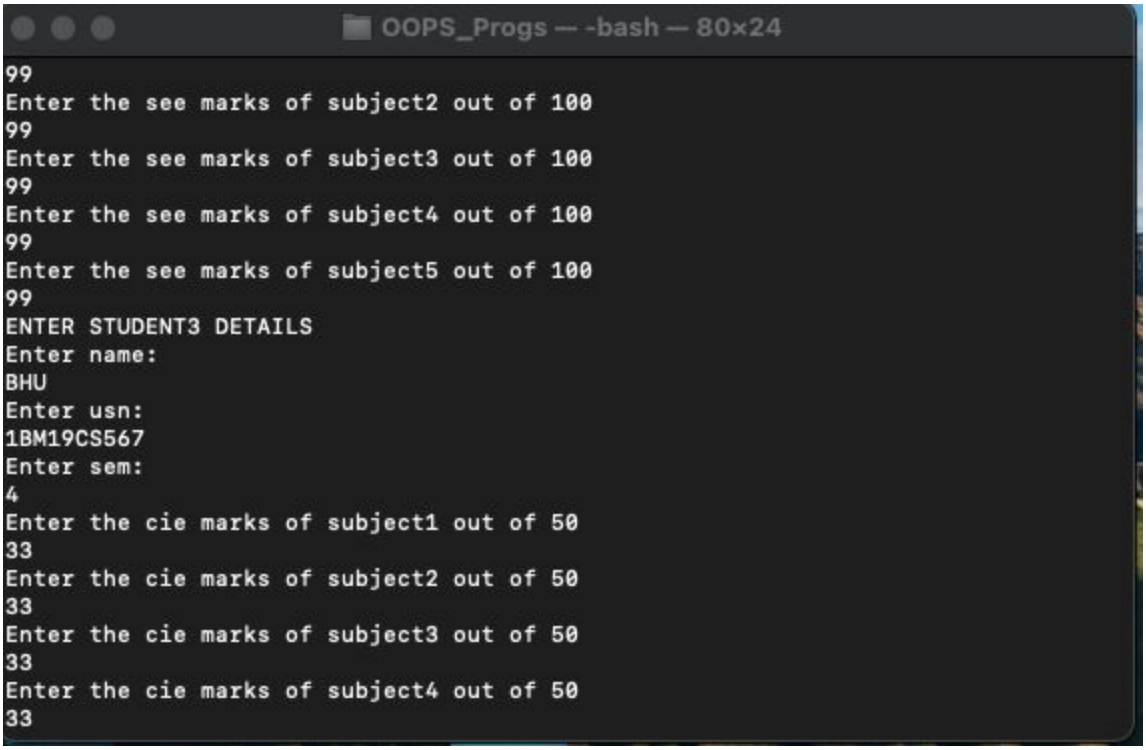
```
        {
            total[j]=ci[i].ciem[j]+(se[i].seem[j]/2);
            System.out.println("Total marks in subject"+(j+1)+" is "+total[j]);
        }
    }
  }
}
```

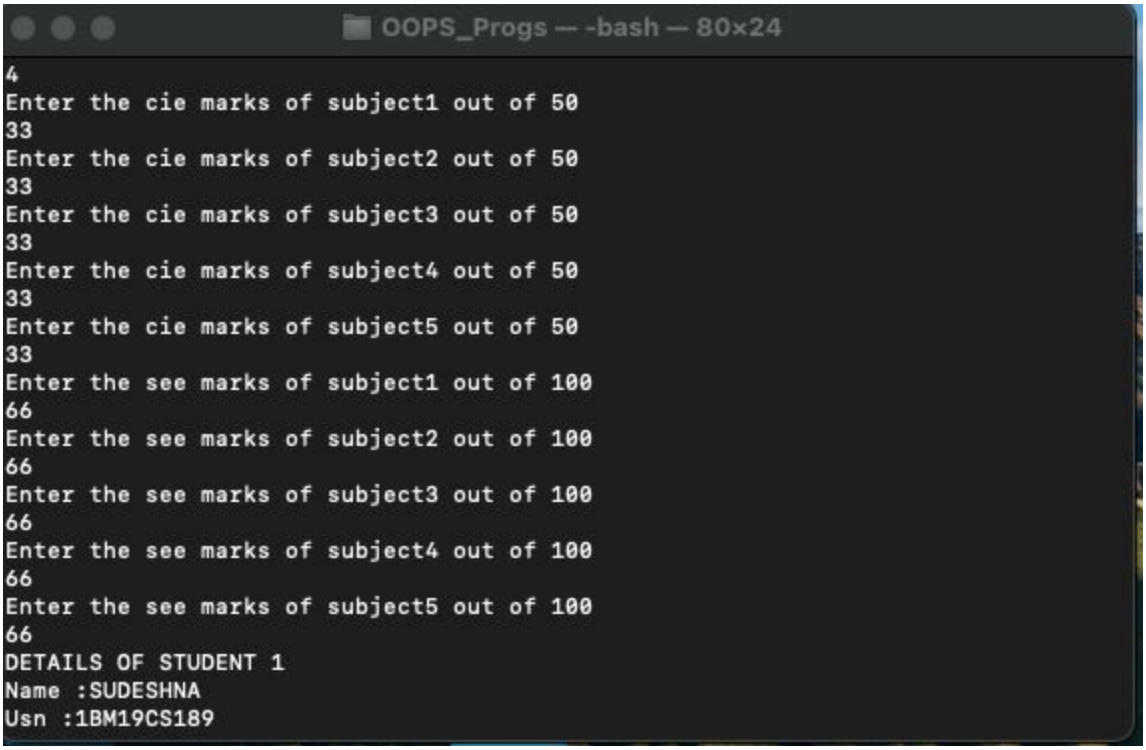
OUTPUT

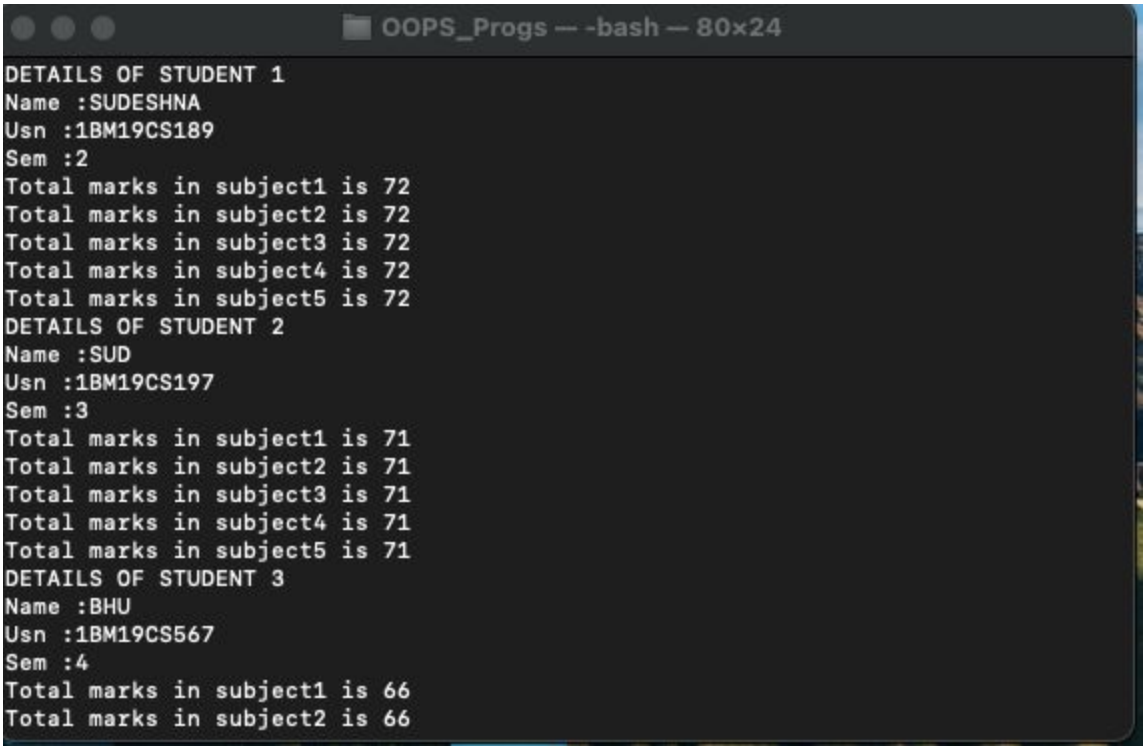



OOPS_Progs — -bash — 80×24
2
Enter the cie marks of subject1 out of 50
34
Enter the cie marks of subject2 out of 50
34
Enter the cie marks of subject3 out of 50
34
Enter the cie marks of subject4 out of 50
34
Enter the cie marks of subject5 out of 50
34
Enter the see marks of subject1 out of 100
77
Enter the see marks of subject2 out of 100
77
Enter the see marks of subject3 out of 100
77
Enter the see marks of subject4 out of 100
77
Enter the see marks of subject5 out of 100
77
ENTER STUDENT2 DETAILS
Enter name:
SUD


OOPS_Progs — -bash — 80×24
ENTER STUDENT2 DETAILS
Enter name:
SUD
Enter usn:
1BM19CS197
Enter sem:
3
Enter the cie marks of subject1 out of 50
22
Enter the cie marks of subject2 out of 50
22
Enter the cie marks of subject3 out of 50
22
Enter the cie marks of subject4 out of 50
22
Enter the cie marks of subject5 out of 50
22
Enter the see marks of subject1 out of 100
99
Enter the see marks of subject2 out of 100
99
Enter the see marks of subject3 out of 100
99
Enter the see marks of subject4 out of 100


OOPS_Progs — -bash — 80×24
99
Enter the see marks of subject2 out of 100
99
Enter the see marks of subject3 out of 100
99
Enter the see marks of subject4 out of 100
99
Enter the see marks of subject5 out of 100
99
ENTER STUDENT3 DETAILS
Enter name:
BHU
Enter usn:
1BM19CS567
Enter sem:
4
Enter the cie marks of subject1 out of 50
33
Enter the cie marks of subject2 out of 50
33
Enter the cie marks of subject3 out of 50
33
Enter the cie marks of subject4 out of 50
33


OOPS_Progs — -bash — 80×24
4
Enter the cie marks of subject1 out of 50
33
Enter the cie marks of subject2 out of 50
33
Enter the cie marks of subject3 out of 50
33
Enter the cie marks of subject4 out of 50
33
Enter the cie marks of subject5 out of 50
33
Enter the see marks of subject1 out of 100
66
Enter the see marks of subject2 out of 100
66
Enter the see marks of subject3 out of 100
66
Enter the see marks of subject4 out of 100
66
Enter the see marks of subject5 out of 100
66
DETAILS OF STUDENT 1
Name :SUDESHNA
Usn :1BM19CS189



LAB PROGRAM 7

QUESTION

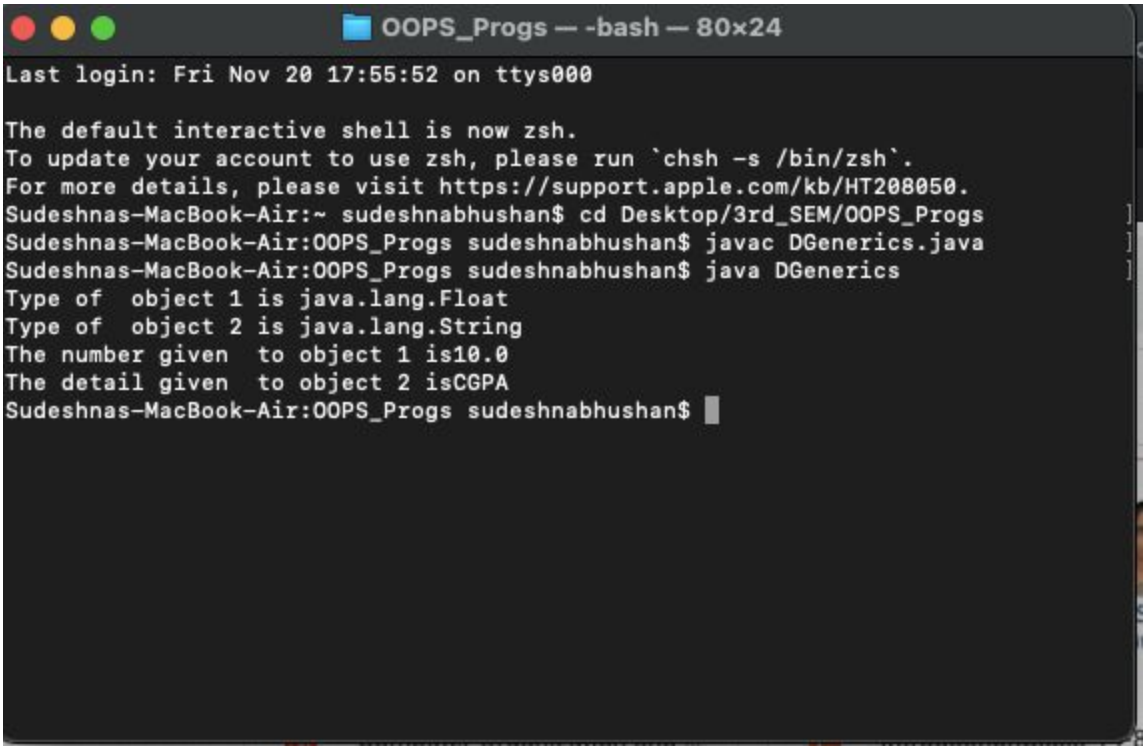
Write a program to demonstrate generics with multiple object parameters.

SOLUTION

```
class GENERICSC<F,S>
{
F object1;
S object2;

GENERICSC(F O1,S O2)
{
object1=O1;
```

```
object2=O2;
}
void printname()
{
System.out.println("Type of  object 1 is "+object1.getClass().getName());
System.out.println("Type of  object 2 is "+object2.getClass().getName());
}
F getob1()
{
return object1;
}
S getob2()
{
return object2;
}
}
public class DGenerics
{
public static void main(String args[])
{
GENERICSC<Float,String> G1=new GENERICSC<Float,String>(10f,"CGPA");
G1.printname();
float FL=G1.getob1();
System.out.println("The number given  to object 1 is"+FL);
String ST=G1.getob2();
System.out.println("The detail given  to object 2 is"+ST);
}
}
```

OUTPUT



LAB PROGRAM 8

QUESTION

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.

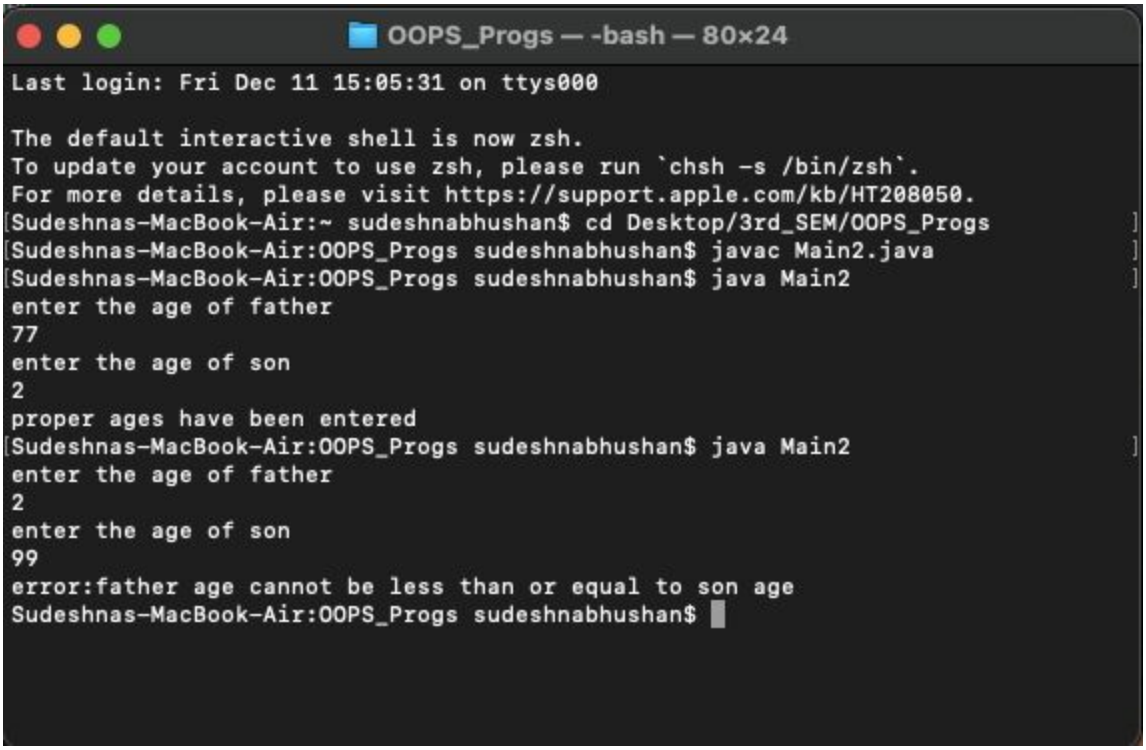
SOLUTION

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

```
class WrongAge extends Exception
{
        private int a1,b1;
        WrongAge(int a,int b)
        {
                a1=a;
                b1=b;
        }
        public String toString()
        {
      if(a1<0||b1<0)
      return "input age cannot be less than 0";
      else if(a1<=b1)
         return "father age cannot be less than or equal to son age ";
      return "";
         }
}

class Father
{
        int fage,sage;
        Scanner sc=new Scanner(System.in);
   Father() throws WrongAge
   {
        System.out.println("enter the age of father");
        fage=sc.nextInt();
        System.out.println("enter the age of son");
        sage=sc.nextInt();
        if(fage<0||sage<0)
                throw new WrongAge(fage,sage);



   }
}

class Son extends Father
{

   Son() throws WrongAge
   {

      if(sage>=fage)
                throw new WrongAge(fage,sage);
```

```
        else
        System.out.println("proper ages have been entered");
    }
}

class Main2
{
        public static void main(String args[])
        {

                try
                {
        Son s=new Son();
                }catch(WrongAge e){
                        System.out.println("error:"+e);

                }
        }
}
```

OUTPUT



LAB PROGRAM 9

QUESTION

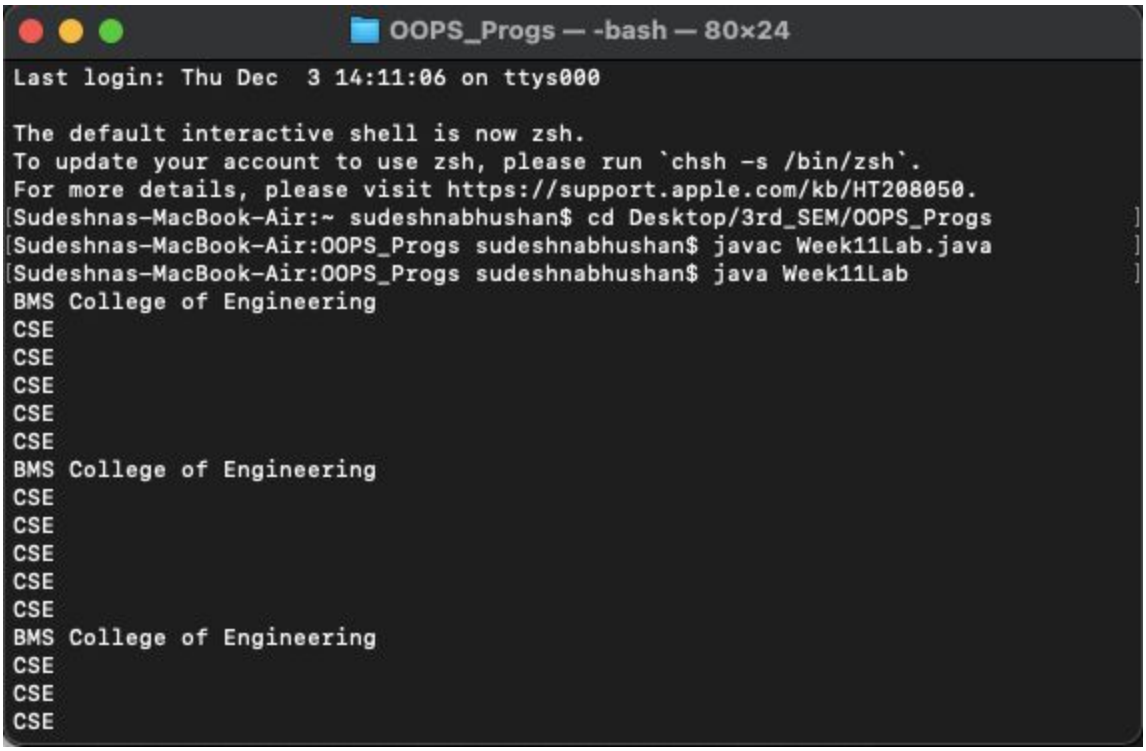
Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

SOLUTION

```
class NewThread implements Runnable
{ Thread t;
   NewThread()
   {
     t = new Thread(this, "NThread");
           t.start();
   }

  public void run()
   {
            try
            {
                     for(int n=100;n>0;n--)
                     {
                              System.out.println("CSE");
                              Thread.sleep(2000);
                     }
            }
            catch(InterruptedException ie)
            {
                     System.out.println("Child Thread Interrupted");
            }

   }
}

class Week11Lab
{
          public static void main(String ss[])
          {
```

```
                NewThread n1=new NewThread();
                //n1.t.start();


                try
                {
                        for(int n=5;n>0;n--)
                        {
                                System.out.println("BMS College of Engineering");
                                Thread.sleep(10000);
                        }
                }
                catch(InterruptedException ie)
                {
                        System.out.println("Main Thread interrupted");
                }

        }
}
```

OUTPUT



LAB PROGRAM 10

QUESTION

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.

SOLUTION

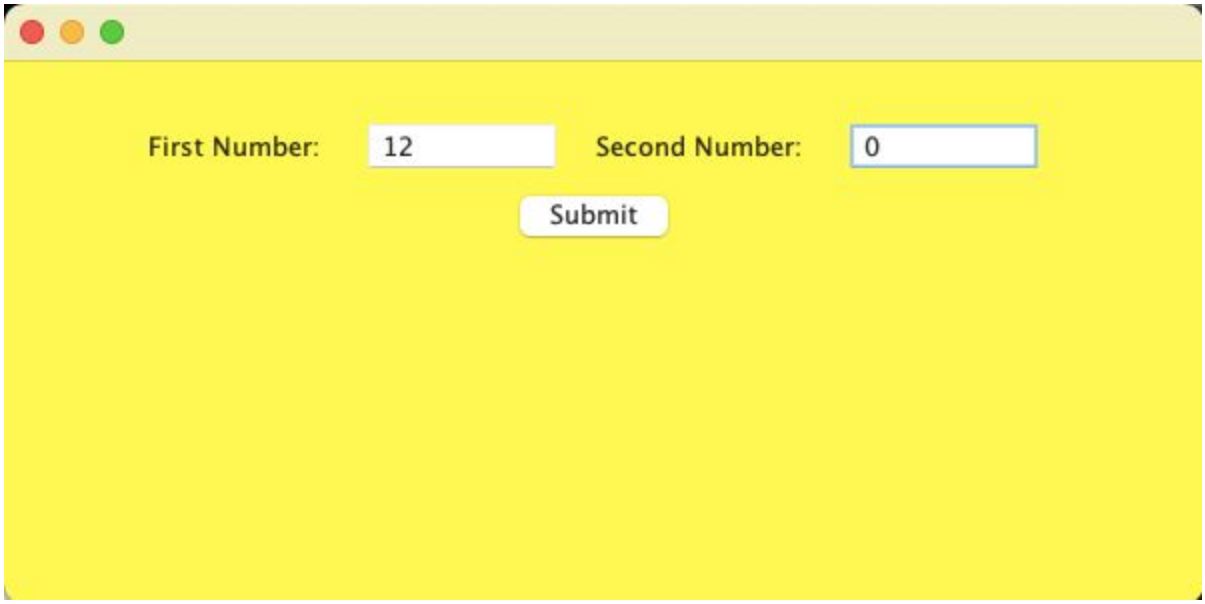
```
import java.awt.*;
import java.awt.event.*;

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);
```

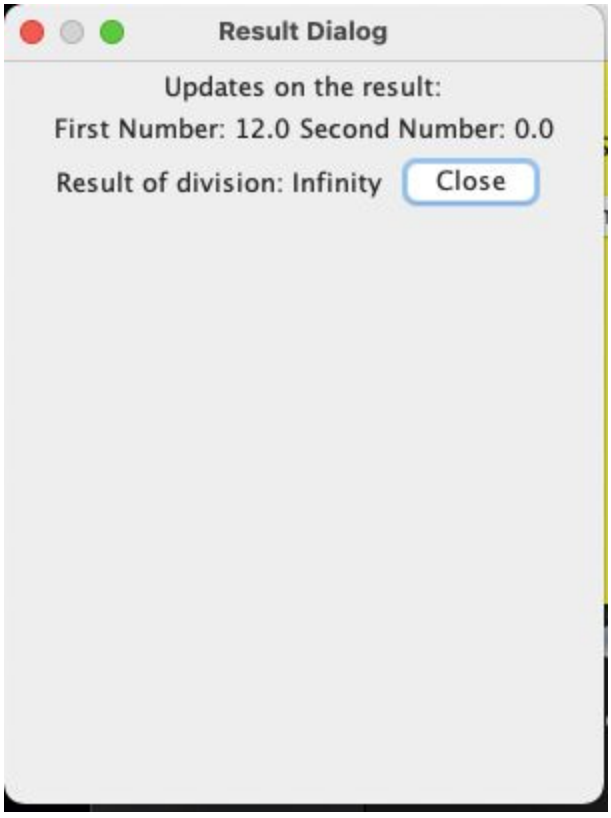
```
		//the following command will make sure that the input char is not visible to the
user
		//(it has been added just to demonstrate). Can be used for passwords.
		//t1.setEchoChar('*');
		//t2.setEchoChar('#');
		this.add(btn,BorderLayout.CENTER);
		this.setVisible(true);
		this.setSize(600, 300);
		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";
```

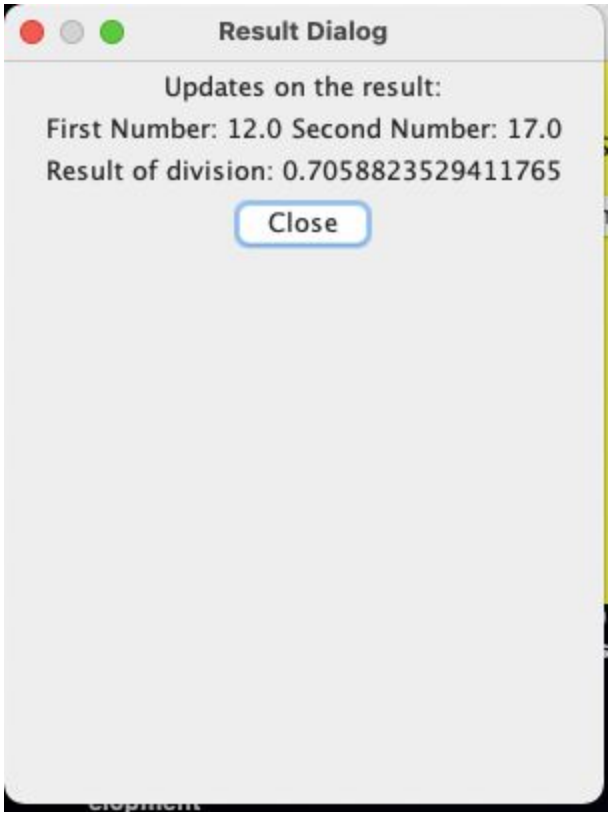
```
                        }
                }
                new MyDialog(this,"Result Dialog",false,msg,n1,n2);
        }
        public static void main(String[] args) {
                new Lab10();
        }
}

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();

        }


}
```

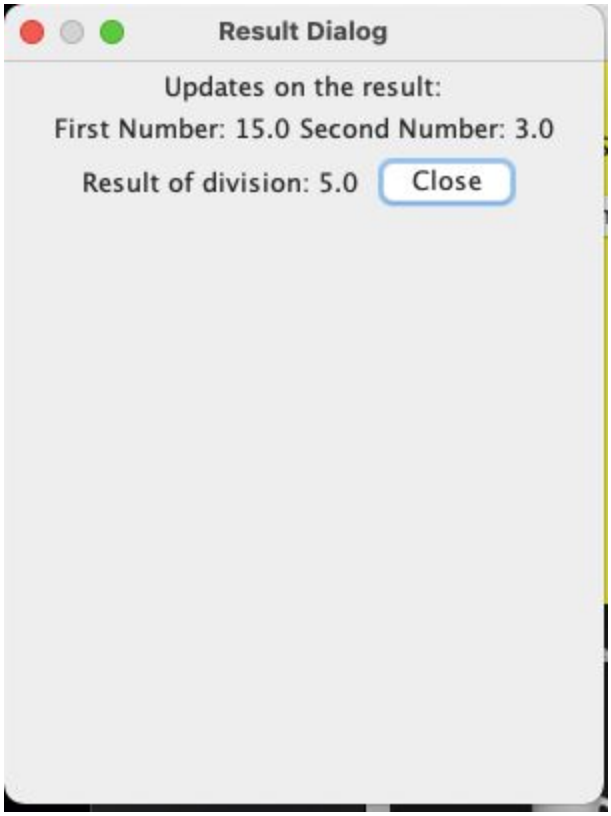
```
class MyWindow extends WindowAdapter{
        public void windowClosing(WindowEvent e) {
                System.exit(0);
        }
}
```

OUTPUT


First Number:
12
Second Number:
0
Submit


Result Dialog
Updates on the result:
First Number: 12.0 Second Number: 0.0
Result of division: Infinity
Close


Result Dialog
Updates on the result:
First Number: 12.0 Second Number: 17.0
Result of division: 0.7058823529411765
Close


Result Dialog
Updates on the result:
First Number: 15.0 Second Number: 3.0
Result of division: 5.0
Close