VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

SNEHA SANTHOSH BHAT (1BM21CS213)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Oct 2022-Feb 2023

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**OBJECT ORIENTED JAVA PROGRAMMING**" carried out by **SNEHA SANTHOSH BHAT (1BM21CS213),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of Object Oriented Java Programming Lab **- (21CS3PCOOJ)** work prescribed for the said degree.

Prameetha Pai
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index Sheet

6	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.	33-37
7	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.	38-41
8	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.	42-46
9	Demonstrate Inter process Communication and deadlock	47-50

Course Outcome

CO1	Apply the knowledge of Java concepts to find the solution for a given problem.
CO2	Analyse the given Java application for correctness/functionalities.
CO3	Develop Java programs / applications for a given requirement.
CO4	Conduct practical experiments for demonstrating features of Java.

LAB PROGRAM 1:

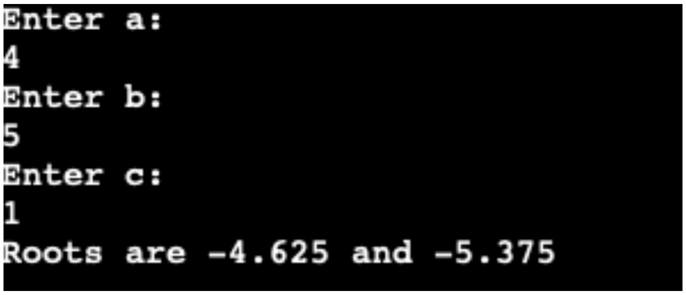
Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions.

Code:

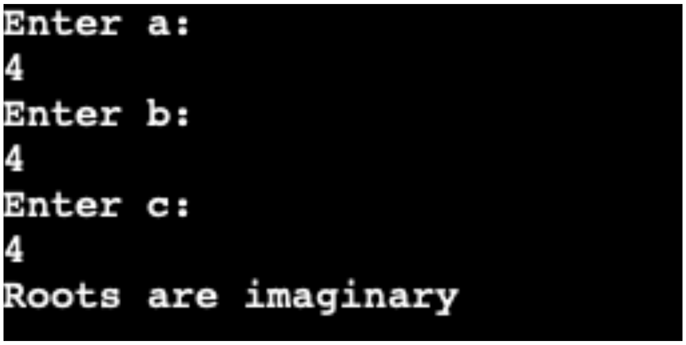
```
import java.util.*;
import java.math.*;
class Quadratic
{
        public static void main(String args[])
        {
                Scanner in=new Scanner(System.in);
                System.out.println("Enter a:");
                double a=in.nextDouble();
                System.out.println("Enter b:");
                double b=in.nextDouble();
                System.out.println("Enter c:");
                double c=in.nextDouble();
                double d=b*b-(4*a*c);
                if(a!=0)
                {
                        if(d>0.0)
```

```
            {
                double r1=(-b+Math.pow(d,0.5)/(2.0*a));
                double r2=(-b-Math.pow(d,0.5)/(2.0*a));
                System.out.println("Roots are "+r1+" and "+r2);
            }
            else if(d==0.0)
            {
                double r1=-b/(2.0*a);
                System.out.println("Root is "+r1);
            }
            else
            {
                System.out.println("Roots are imaginary");
            }
        }
    }
}
```

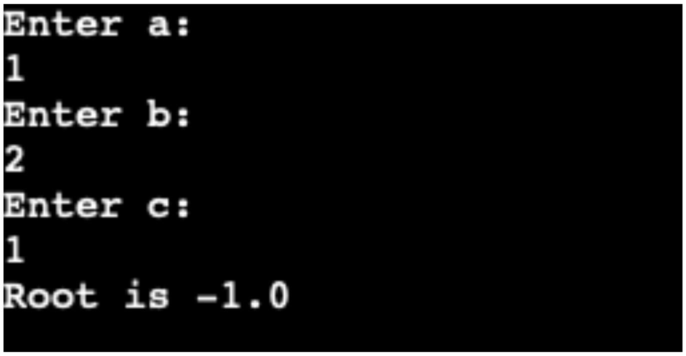
Output:



```
Enter a:
4
Enter b:
5
Enter c:
1
Roots are -4.625 and -5.375
```



```
Enter a:
4
Enter b:
4
Enter c:
4
Roots are imaginary
```



```
Enter a:
1
Enter b:
2
Enter c:
1
Root is -1.0
```

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.

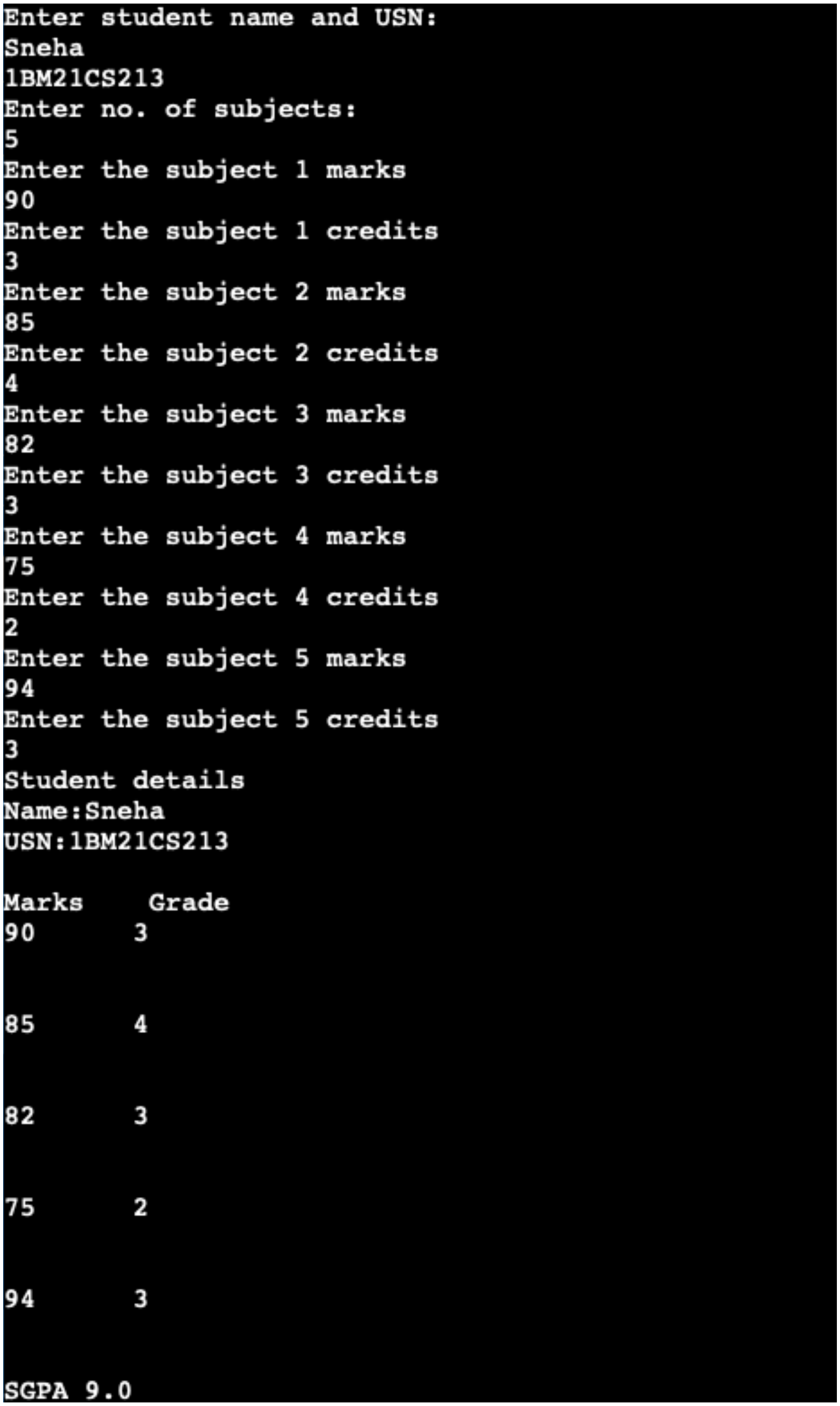
Code:

```
import java.util.*;

class student

{

        String usn,name;

        int credits[]=new int[50];

        int marks[]=new int[50];

        int gradepoint[]=new int[50];

        int n,num=0,denum=0;

        double sgpa; int i;

        void accept()

        {

                Scanner in=new Scanner(System.in);

                System.out.println("Enter student name and USN:");

                name=in.next();
```

```
    usn=in.next();
    System.out.println("Enter no. of subjects:");
    n=in.nextInt();
    for(i=0;i<n;i++)
    {
        System.out.println("Enter the subject "+(i+1)+" marks");
        marks[i]=in.nextInt();
        System.out.println("Enter the subject "+(i+1)+" credits");
        credits[i]=in.nextInt();
    }
}
void calculate()
{
    for(i=0;i<n;i++)
    {
        if(marks[i]>=90)
            gradepoint[i]=10;
        else if((marks[i]>=80)&&(marks[i]<90))
            gradepoint[i]=9;
        else if((marks[i]>=70)&&(marks[i]<80))
```

```
                        gradepoint[i]=8;
                else if((marks[i]>=60)&&(marks[i]<70))
                        gradepoint[i]=7;
                else if((marks[i]>=50)&&(marks[i]<60))
                        gradepoint[i]=6;
                else if((marks[i]>=40)&&(marks[i]<50))
                        gradepoint[i]=4;
                else
                        gradepoint[i]=0;
                num+=credits[i]*gradepoint[i];
                denum+=credits[i];
        }
        sgpa=num/denum;
}
void display()
{
        System.out.println("Student details");
        System.out.println("Name:"+name+"\n"+"USN:"+usn+"\n");
        System.out.println("Marks\t Grade");
        for(i=0;i<n;i++)
        {
            System.out.println(marks[i]+"\t"+credits[i]);
            System.out.println("\n");
        }
        System.out.println("SGPA "+sgpa);
    }
}
class studentdemo
{
    public static void main(String args[])
    {
        student ob=new student();
        ob.accept();
        ob.calculate();
        ob.display();
    }
}
```

Output:



```
Enter student name and USN:
Sneha
1BM21CS213
Enter no. of subjects:
5
Enter the subject 1 marks
90
Enter the subject 1 credits
3
Enter the subject 2 marks
85
Enter the subject 2 credits
4
Enter the subject 3 marks
82
Enter the subject 3 credits
3
Enter the subject 4 marks
75
Enter the subject 4 credits
2
Enter the subject 5 marks
94
Enter the subject 5 credits
3
Student details
Name:Sneha
USN:1BM21CS213

Marks    Grade
90      3


85      4


82      3


75      2


94      3


SGPA 9.0
```

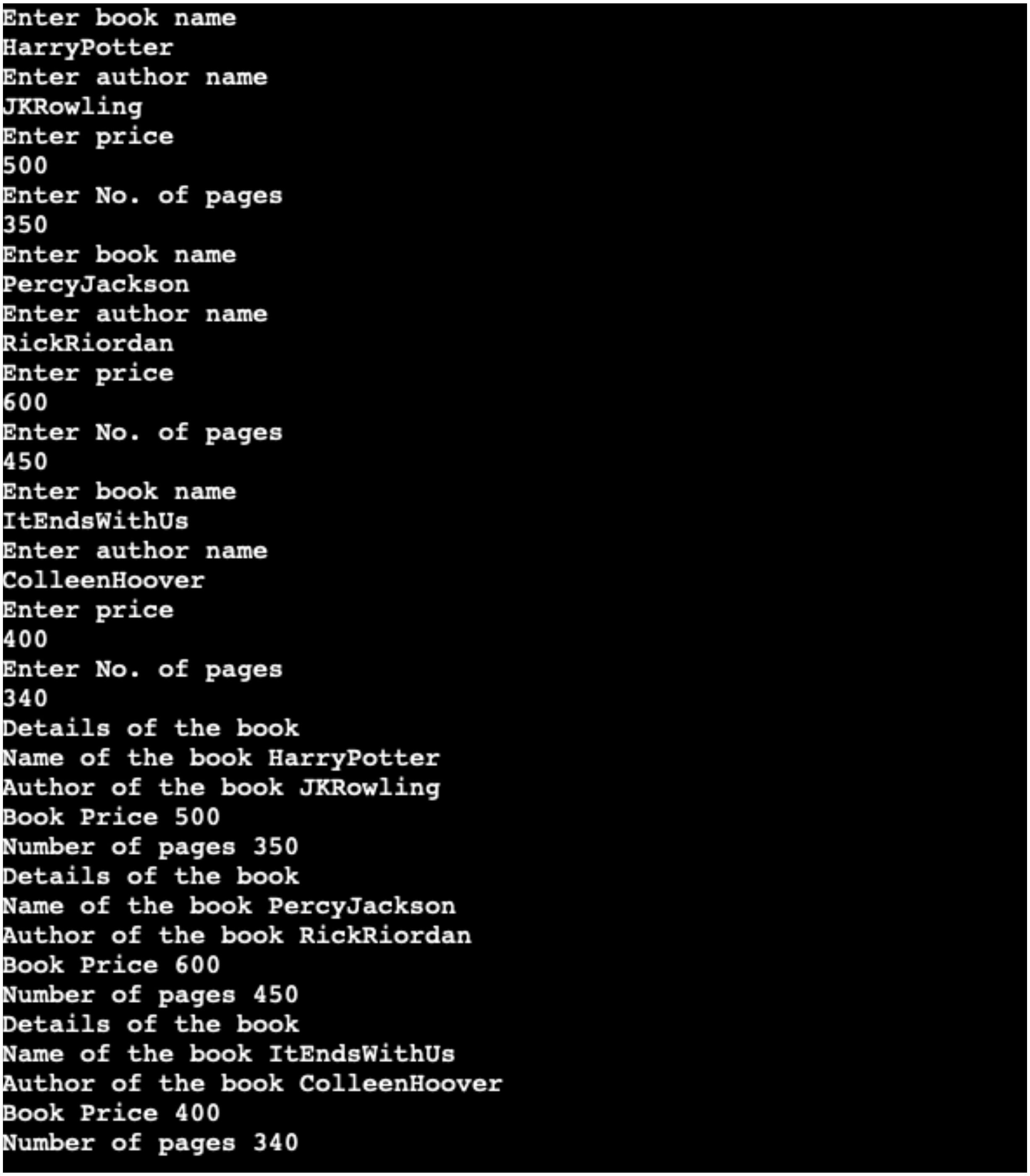
LAB PROGRAM 3:

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.

Code:

```
import java.util.*;

import java.lang.*;

class Book

{

        String name,author;

        int price,num_pages;

        void getval()

        {

                Scanner bk=new Scanner(System.in);

                System.out.println("Enter book name");

                name=bk.next();

                System.out.println("Enter author name");
                author=bk.next();

                System.out.println("Enter price ");

                price=bk.nextInt();

                System.out.println("Enter No. of pages");

                num_pages=bk.nextInt();

        }

        void display()

        {

                System.out.println("Details of the book");

                System.out.println("Name of the book"+" "+name);

                System.out.println("Author of the book"+" "+author);

                System.out.println("Book Price"+" "+price);

                System.out.println("Number of pages"+" "+num_pages);

        }

}

class Book1

{

        public static void main(String args[])

        {

                Book b[]=new Book[3];
                for(int i=0;i<3;i++)

                {

                        b[i]=new Book();

                }

                for(int j=0;j<3;j++)

                {

                        b[j].getval();

                }

                for(int j=0;j<3;j++)

                {

                        b[j].display();

                }

        }

}
```

Output:



```
Enter book name
HarryPotter
Enter author name
JKRowling
Enter price
500
Enter No. of pages
350
Enter book name
PercyJackson
Enter author name
RickRiordan
Enter price
600
Enter No. of pages
450
Enter book name
ItEndsWithUs
Enter author name
ColleenHoover
Enter price
400
Enter No. of pages
340
Details of the book
Name of the book HarryPotter
Author of the book JKRowling
Book Price 500
Number of pages 350
Details of the book
Name of the book PercyJackson
Author of the book RickRiordan
Book Price 600
Number of pages 450
Details of the book
Name of the book ItEndsWithUs
Author of the book ColleenHoover
Book Price 400
Number of pages 340
```

LAB PROGRAM 4:

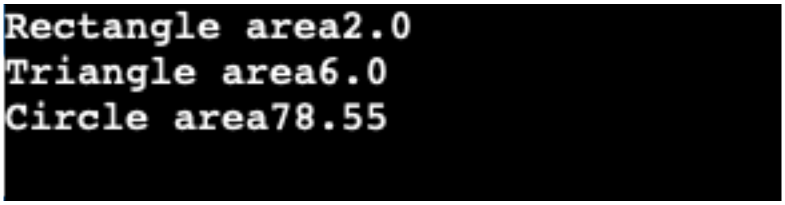
Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea(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

Code:

```
import java.lang.*;

abstract class shape

{

        int a,b;

        double area;

        final double pi=3.142;

        shape(int x,int y)

        {

                a=x;

                b=y;

                area=0;

        }
        abstract void printArea();
}
class Rectangle extends shape
{
        Rectangle(int a,int b)
        {
                super(a,b);
        }
        void printArea()
        {
                area=a*b;
                System.out.println("Rectangle area"+area);
        }
}
class triangle extends shape
{
        triangle(int a,int b)
        {
        super(a,b);
        }
```

```
    void printArea()
    {
        area=(a*b)*(0.5);
        System.out.println("Triangle area"+area);
    }
}
class circle extends shape
{
    circle(int a)
    {
        super(a,-1);
    }
    void printArea()
    {
        area=pi*Math.pow(a,2);
        System.out.println("Circle area"+area);
    }
}
class demoshape
{
    public static void main(String args[])
    {
        Rectangle r1=new Rectangle(1,2);
        triangle t1=new triangle(3,4);
        circle c1=new circle(5);
        shape ref;
        ref=r1;
        ref.printArea();
        ref=t1;
        ref.printArea();
        ref=c1;
        ref.printArea();
    }
}
```

Output:



```
Rectangle area2.0
Triangle area6.0
Circle area78.55
```

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 Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

Code:

```
import java.util.Scanner;
import java.lang.Math;
class Account
{
	String name, acc_type;
	int acc_no;
	double bal,dep;
	Scanner scan= new Scanner(System.in);
	void setd()
	{
		System.out.println("Enter your Name:");
		name=scan.next();
		System.out.println("Enter your Account Number:");
		acc_no=scan.nextInt();
		System.out.println("Enter your Account type: (Savings/Current)");
		acc_type=scan.next();
		System.out.println("Enter the Bank Balance:");
		bal=scan.nextInt();
	}
```

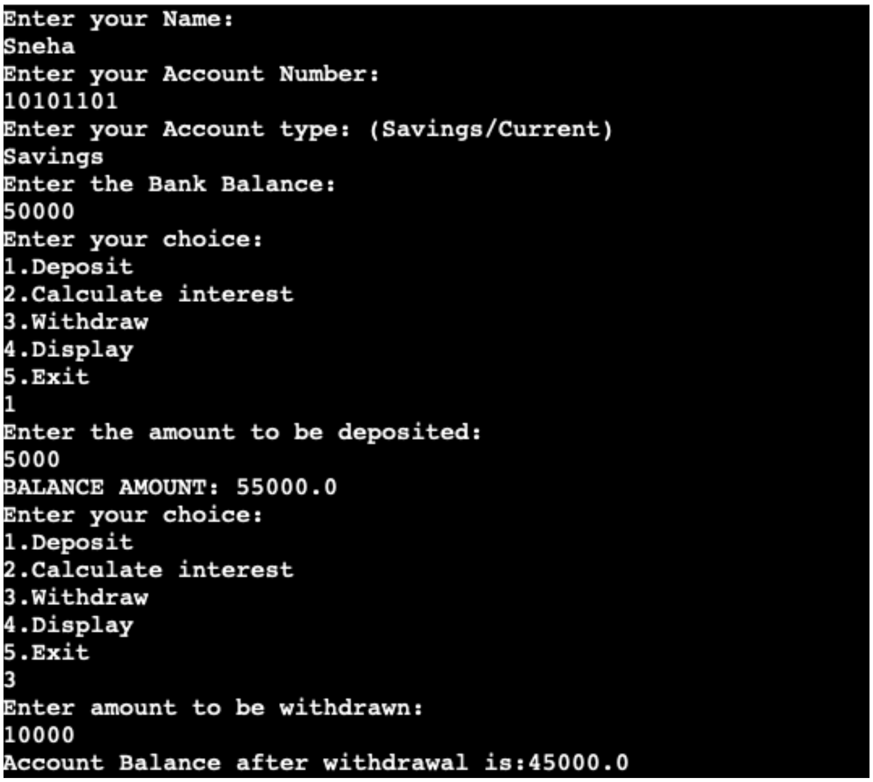
```
        void disp()
        {
                System.out.println("Name: "+name);
                System.out.println("Account Number: "+acc_no);
                System.out.println("Account Type: "+acc_type);
                System.out.println("Current balance is: "+bal);
        }
        void deposit()
        {
                System.out.println("Enter the amount to be deposited:");
                dep=scan.nextInt();
                bal+=dep;
                System.out.println("BALANCE AMOUNT: "+bal);
        }
}
class Cur_acct extends Account
{
        int penal()
        {
                double min, pen;
                System.out.println("Enter Minimum balance & penalty amount if not
        followed:");
                min=5000; pen=min*0.05;
                if(bal<min)
                {
                        bal-=pen;
                        System.out.println("Penalty imposed for having insufficient
                balance"); return 0;
                }
                else
                        {System.out.println("No penalty");
                          return 1;}
        }
        void withdrawal()
        {
                double amt;
                System.out.println("Enter amount to be withdrawn:");
                amt=scan.nextInt();
                 int a= penal();
                if(a==1)
                {
                if(bal>=amt)
                {
                        bal=bal-amt;
                        System.out.println("Account Balance after withdrawal
                is:" +bal);}
            }
            else
                System.out.println("The amount can't be withdrawn");
        }
}
class Sav_acct extends Account
{
        void calc_interest()
        {
                System.out.println("Enter Time in years and Rate of interest");
                double t=scan.nextDouble(); double r=scan.nextDouble();
                double CI = bal*Math.pow((1 + r/ 100), t);
                System.out.println("ACCOUNT BALANCE:" + bal);
                System.out.println("Compounding interest:"+ CI);
        }
        void withdrawal()
        {
                double amt;
                System.out.println("Enter amount to be withdrawn:");
                amt=scan.nextInt();
                if(bal>=amt)
                { bal=bal-amt;
                System.out.println("Account Balance after withdrawal is:" +bal);}
                else
                        System.out.println("The amount can't be withdrawn");
        }
}
class Bank
{
         public static void main(String arg[])
        {
                Scanner ss=new Scanner(System.in);
                Account b1=new Account();
                b1.setd();
                if(b1.acc_type.equals("Savings"))
```

```
{
    Sav_acct s1=new Sav_acct();
    //s1=b1;
    s1.name=b1.name; s1.acc_no=b1.acc_no;
s1.acc_type=b1.acc_type; s1.bal=b1.bal;
    while(true)
    {
        System.out.println("Enter your
    choice:\n1.Deposit\n2.Calculate
    interest\n3.Withdraw\n4.Display\n5.Exit");
        int choice=ss.nextInt();
        switch(choice)
        {
            case 1: s1.deposit(); break;
            case 2: s1.calc_interest(); break;
            case 3: s1.withdrawal(); break;
            case 4: s1.disp(); break;
            case 5: System.exit(0);
            default: System.out.println("Invalid input");
        }
    }
```

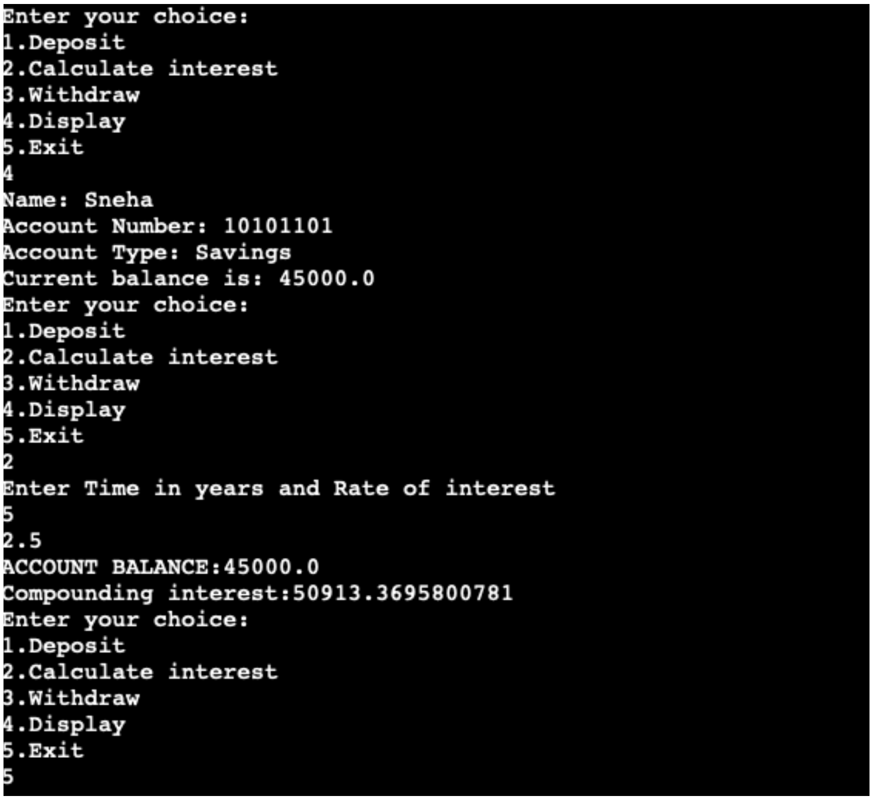
```
}

else if(b1.acc_type.equals("Current"))

{

    Cur_acct c1=new Cur_acct();

    c1.name=b1.name; c1.acc_no=b1.acc_no;
c1.acc_type=b1.acc_type; c1.bal=b1.bal;

    while(true)

    {

        System.out.println("Enter your
    choice:\n1.Deposit\n2.Penalty
    Check\n3.Withdraw\n4.Display\n5.Exit");

        int choice=ss.nextInt();

        switch(choice)

        {

            case 1: c1.deposit(); break;

            case 2: c1.penal(); break;

            case 3: c1.withdrawal(); break;

            case 4: c1.disp(); break;

            case 5: System.exit(0);

            default: System.out.println("Invalid input");

        }
```

```
                    }
                }
                else
                    System.out.println("Invalid Account type");
            }
        }
```

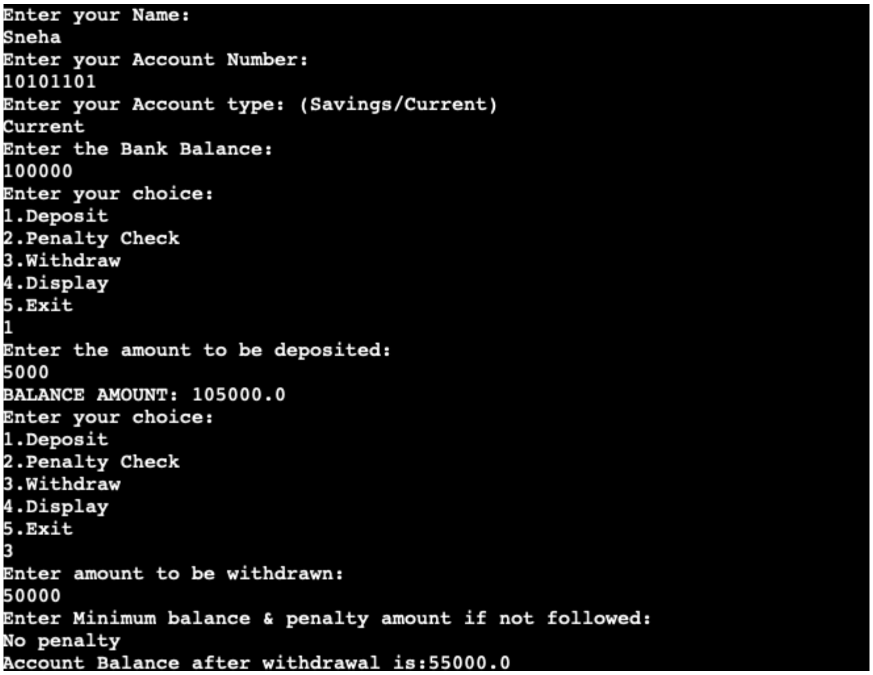
Output:



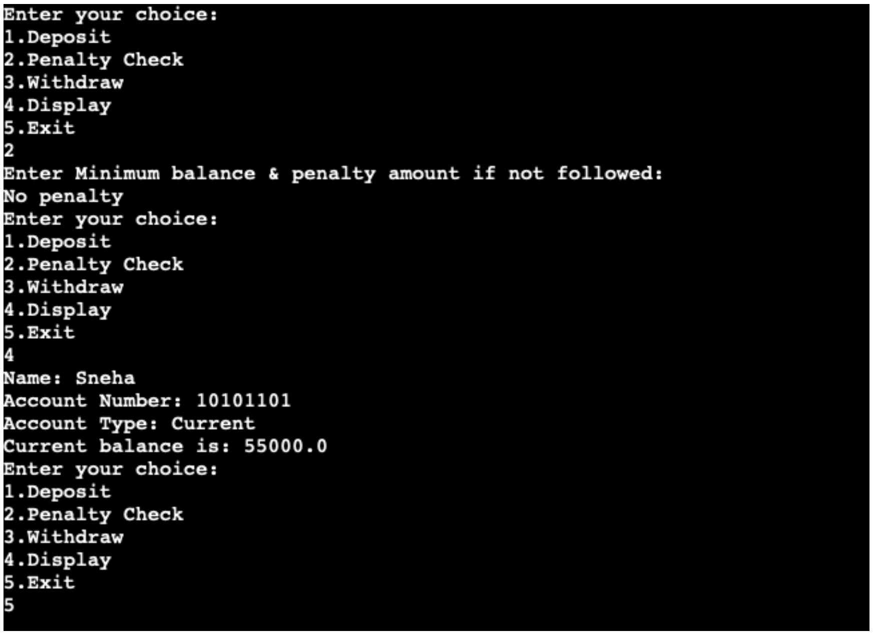
```
Enter your Name:
Sneha
Enter your Account Number:
10101101
Enter your Account type: (Savings/Current)
Savings
Enter the Bank Balance:
50000
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
1
Enter the amount to be deposited:
5000
BALANCE AMOUNT: 55000.0
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
3
Enter amount to be withdrawn:
10000
Account Balance after withdrawal is:45000.0
```



```
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
4
Name: Sneha
Account Number: 10101101
Account Type: Savings
Current balance is: 45000.0
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
2
Enter Time in years and Rate of interest
5
2.5
ACCOUNT BALANCE:45000.0
Compounding interest:50913.3695800781
Enter your choice:
1.Deposit
2.Calculate interest
3.Withdraw
4.Display
5.Exit
5
```



```
Enter your Name:
Sneha
Enter your Account Number:
10101101
Enter your Account type: (Savings/Current)
Current
Enter the Bank Balance:
100000
Enter your choice:
1.Deposit
2.Penalty Check
3.Withdraw
4.Display
5.Exit
1
Enter the amount to be deposited:
5000
BALANCE AMOUNT: 105000.0
Enter your choice:
1.Deposit
2.Penalty Check
3.Withdraw
4.Display
5.Exit
3
Enter amount to be withdrawn:
50000
Enter Minimum balance & penalty amount if not followed:
No penalty
Account Balance after withdrawal is:55000.0
```



```
Enter your choice:
1.Deposit
2.Penalty Check
3.Withdraw
4.Display
5.Exit
2
Enter Minimum balance & penalty amount if not followed:
No penalty
Enter your choice:
1.Deposit
2.Penalty Check
3.Withdraw
4.Display
5.Exit
4
Name: Sneha
Account Number: 10101101
Account Type: Current
Current balance is: 55000.0
Enter your choice:
1.Deposit
2.Penalty Check
3.Withdraw
4.Display
5.Exit
5
```

LAB PROGRAM 6:

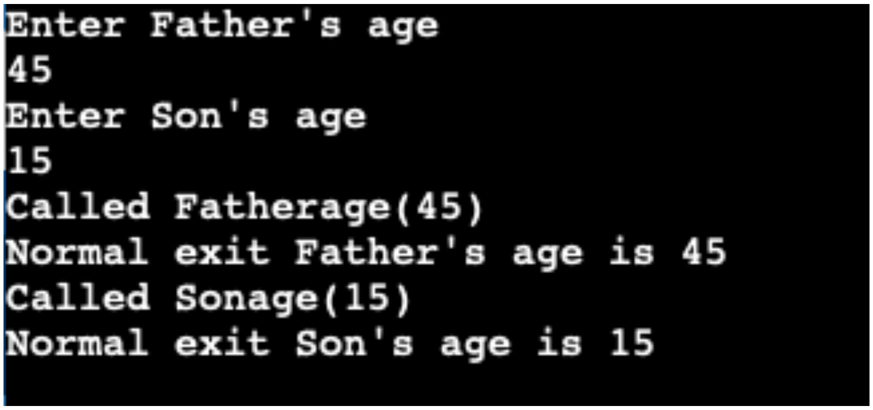
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son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

Code:

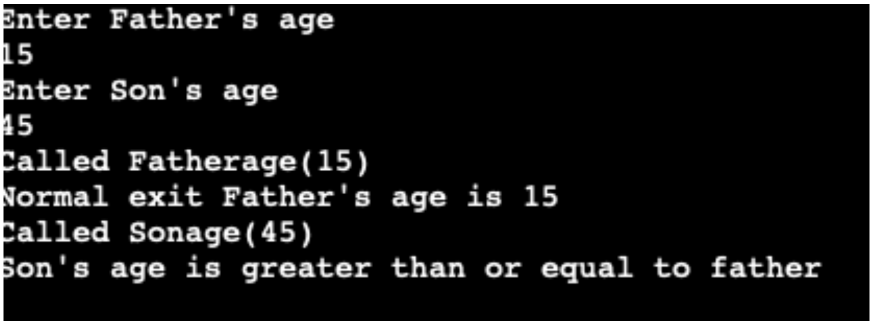
```
import java.util.Scanner;
class Father extends Exception
{
        int fage;
        Father(int x)
        {
                fage=x;
        }
        public String toString()
        {
                return "Father's age is wrong";
        }
}
class Son extends Father
{
        int sage;
        Son(int x,int y)
        {
                super(x);
                sage=y;
        }
        public String toString()
        {
                return "Son's age is greater than or equal to father";
        }
}
class Wrongage
{
        static int x,y;
        static void Fatherage(int x) throws Father
        {
```

```
        System.out.println("Called Fatherage("+x+")");

        if(x<0)

        throw new Father(x);

        System.out.println("Normal exit Father's age is "+x);

}

static void Sonage(int x,int y) throws Son

{

        System.out.println("Called Sonage("+y+")");

        if(y>=x)

        throw new Son(x,y);

        System.out.println("Normal exit Son's age is "+y);

}

public static void main(String args[])

{

        Scanner input=new Scanner(System.in);

        System.out.println("Enter Father's age");

        x=input.nextInt();

        System.out.println("Enter Son's age");

        y=input.nextInt();

        try
        {
            Fatherage(x);
        }
        catch(Father e)
        {
            System.out.println(e);
        }
        try
        {
            Sonage(x,y);
        }
        catch(Son e)
        {
            System.out.println(e);
        }
    }
}
```

Output:



```
Enter Father's age
45
Enter Son's age
15
Called Fatherage(45)
Normal exit Father's age is 45
Called Sonage(15)
Normal exit Son's age is 15
```



```
Enter Father's age
15
Enter Son's age
45
Called Fatherage(15)
Normal exit Father's age is 15
Called Sonage(45)
Son's age is greater than or equal to father
```

LAB PROGRAM 7:

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.

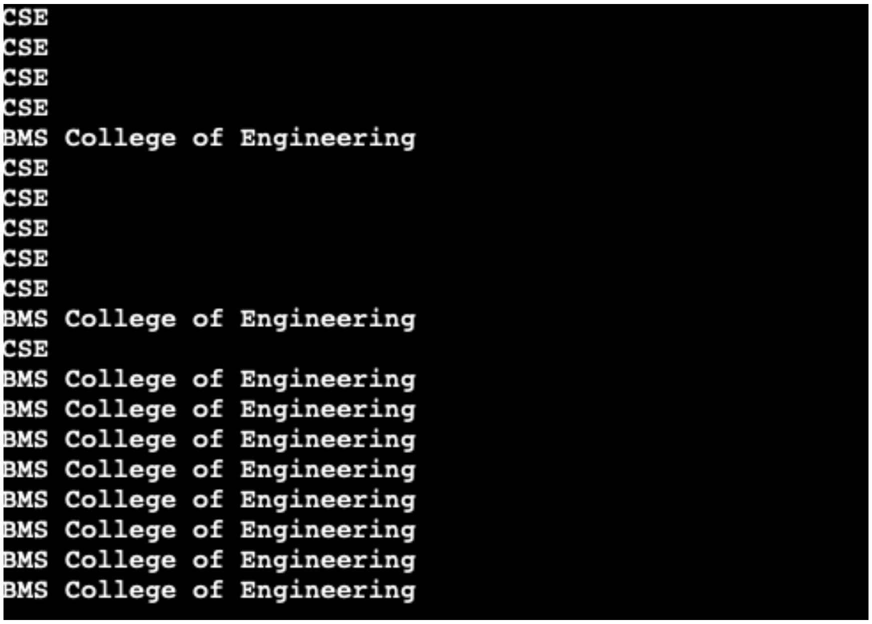
Code:

```
class Thread1 extends Thread
{
        public void run()
        {
                int i=0;
                while(i<10)
                {
                        try
                        {
                                Thread.sleep(10000);
                                System.out.println("BMS College of Engineering");
                        }
                        catch(Exception e)
                        {
```

```
                System.out.println("Exception:"+e);
            }
            i++;
        }
    }
}
class Thread2 extends Thread
{
    public void run()
    {
        int i=0;
        while(i<10)
        {
            try
            {
                Thread.sleep(2000);
                System.out.println("CSE");
            }
            catch(Exception e)
            {
```

```
                                System.out.println("Exception:"+e);

                        }

                        i++;

                }

        }

}

public class Try

{

        public static void main(String args[])

        {

                Thread t1=new Thread1();

                Thread t2=new Thread2();

                t1.start();

                t2.start();

        }

}
```

Output:



```
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
```

LAB PROGRAM 8:

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.

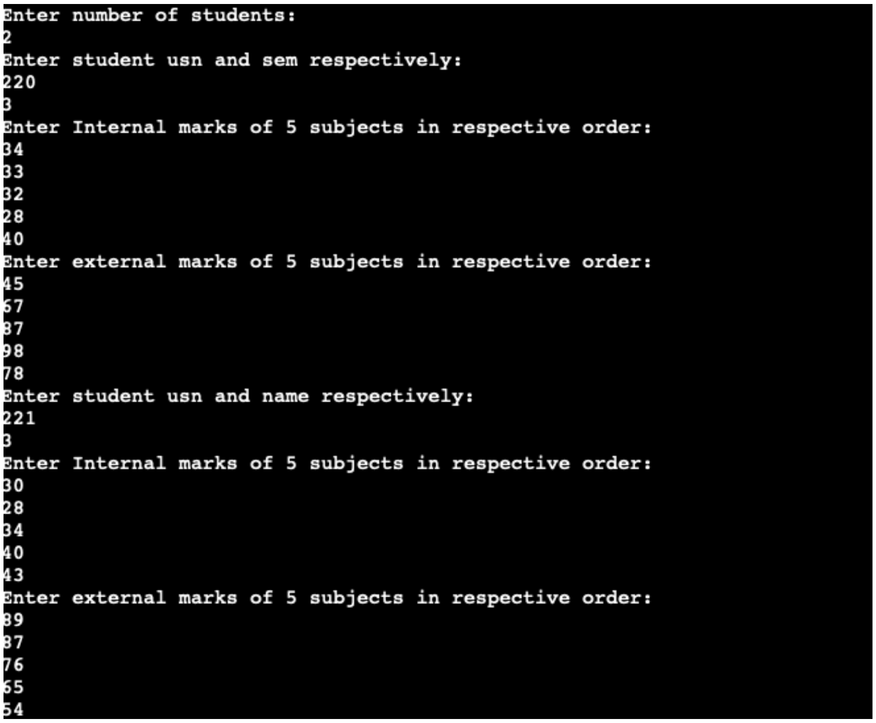
Code:

```
package cie;

public class internals

{

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

}

package cie;

public class student

{

        public String name;

        public int usn;
        public int sem;
}
package see;
import cie.internals;
public class External extends internals
{
        public int external[]=new int[5];
}
import java.util.Scanner;
import cie.student;
import see.External;
public class Marks
{
        public static void main(String args[])
        {
                int n;
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter number of students:");
                n=sc.nextInt();
                External student[]=new External[n];
```

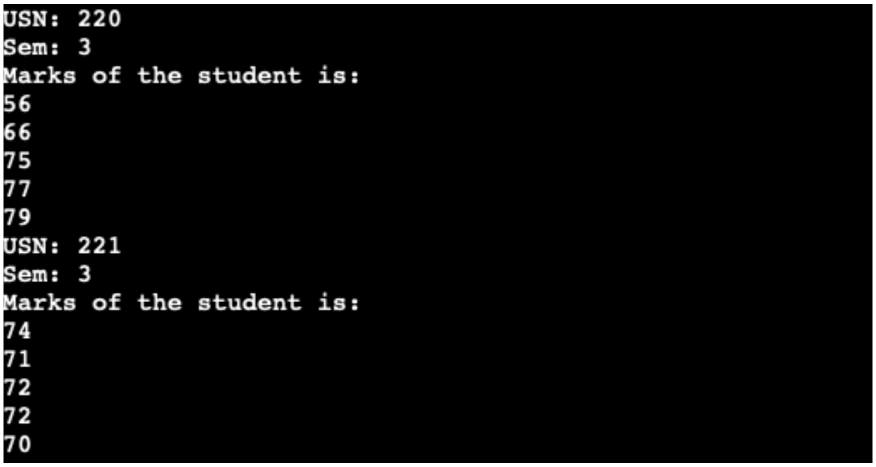
```
student details[]=new student[n];
int final_marks[][]=new int[n][5];
for(int i=0;i<n;i++)
{
	System.out.println("Enter student usn and name respectively:");
	student[i]=new External();
	details[i]=new student();
	System.out.println("Enter Internal marks of 5 subjects in
respective order:");
	for(int j=0;j<5;j++)
	{
		student[i].internal[j]=sc.nextInt();
	}
	System.out.println("Enter external marks of 5 subjects in
respective order:");
	for(int k=0;k<5;k++)
	{
		student[i].external[k]=sc.nextInt();
	}
}
for(int i=0;i<n;i++)
```

```
        {
            for(int j=0;j<5;j++)
            {
                final_marks[i][j]=student[i].internal[j]+(int)(student[i].external[j]/2);
            }
        }
        for(int i=0;i<n;i++)
        {
            System.out.println("USN:"+details[i].usn);
            System.out.println("Sem:"+details[i].sem);
            System.out.println("Marks of the student is:");
            for(int j=0;j<5;j++)
            {
                System.out.println(final_marks[i][j]);
            }
        }
    }
}
```

Output:



```
Enter number of students:
2
Enter student usn and sem respectively:
220
3
Enter Internal marks of 5 subjects in respective order:
34
33
32
28
40
Enter external marks of 5 subjects in respective order:
45
67
87
98
78
Enter student usn and name respectively:
221
3
Enter Internal marks of 5 subjects in respective order:
30
28
34
40
43
Enter external marks of 5 subjects in respective order:
89
87
76
65
54
```



```
USN: 220
Sem: 3
Marks of the student is:
56
66
75
77
79
USN: 221
Sem: 3
Marks of the student is:
74
71
72
72
70
```

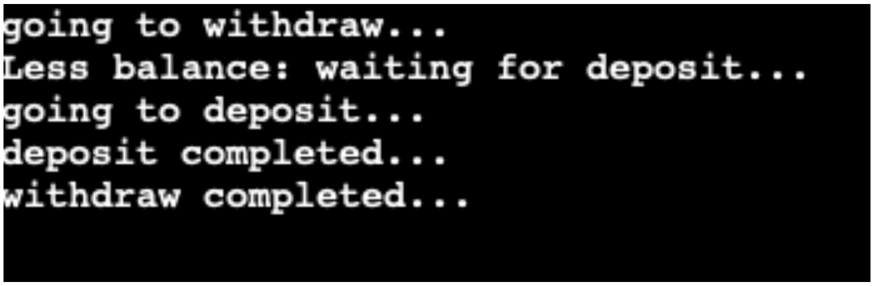
LAB PROGRAM 9:

Demonstrate Inter process Communication and deadlock.

Code:

```
class Customer
{
        int amount=10000;
        synchronized void withdraw(int amount)
        {
                 System.out.println("going to withdraw...");
                if(this.amount<amount)
                {
                        System.out.println("Less balance: waiting for deposit...");
                        try
                        {
                                wait();
                        }
                        catch(Exception e)
                        {   }
                }
        this.amount-=amount;
        System.out.println("withdraw completed...");
    }
    synchronized void deposit(int amount)
    {
        System.out.println("going to deposit...");
        this.amount+=amount;
        System.out.println("deposit completed...");
        notify();
    }
}
class Test
{
    public static void main (String[] args)
    {
        final Customer c=new Customer();
        new Thread()
        {
            public void run()
            {
                c.withdraw(15000);
            }
        }.start();
        new Thread()
        {
            public void run()
            {
                c.deposit(10000);
            }
        }.start();
    }
}
```

Output:



```
going to withdraw...
Less balance: waiting for deposit...
going to deposit...
deposit completed...
withdraw completed...
```