VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

SAREDDY POOJYA
SREE(1BM23CS303)

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 Sep
2024-Jan 2025

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 **SAREDDY POOJYA SREE(1BM23CS303),** 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 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Java Programming Lab - (23CS3PCOOJ)** work prescribed for the said degree.

Dr. Nandhini Vineeth

Associate Professor,
Department of CSE,
BMSCE, Bengaluru

Dr. Kavitha Sooda

Professor and Head,
Department of CSE
BMSCE, Bengaluru

INDEX

LAB PROGRAM -1

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 quadratic formula. If the discriminate is negative, display a message stating, there are no real solutions.

```
import java.util.Scanner;
public class main{
{
  public static void main (String[] args)
  {
    int d; int a, b, c;
    double r1, r2;
    Scanner ss = new Scanner(System.in);
    System.out.println("a,b,c:");
    a = ss.nextInt();
    b = ss.nextInt();
    c = ss.nextInt();
    d = b*b - 4*a*c;
if (d>0)
{
   r1 = (-b + Math.pow(d,0.5))/(2*a);
   r2 = (-b - Math.pow(d,0.5))/(2*a);
  System.out.println(" R1 = "+ r1 + ", R2 = " +r2);
}
 else if (d==0)
  {
    r1 = r2 = -b/(2*a);
  System.out.println(" R1 = R2 = " + r1);
  }
 else
  {
   System.out.println("There are no real solutions");
  }
}
```

Output: a, b, c:

1

-5

6

Roots R1 = 3.0 , R2 = 2.0

Output: a, b, c:

1

-4

4

R1 = R2 = 2.0

Output : a, b, c:

1

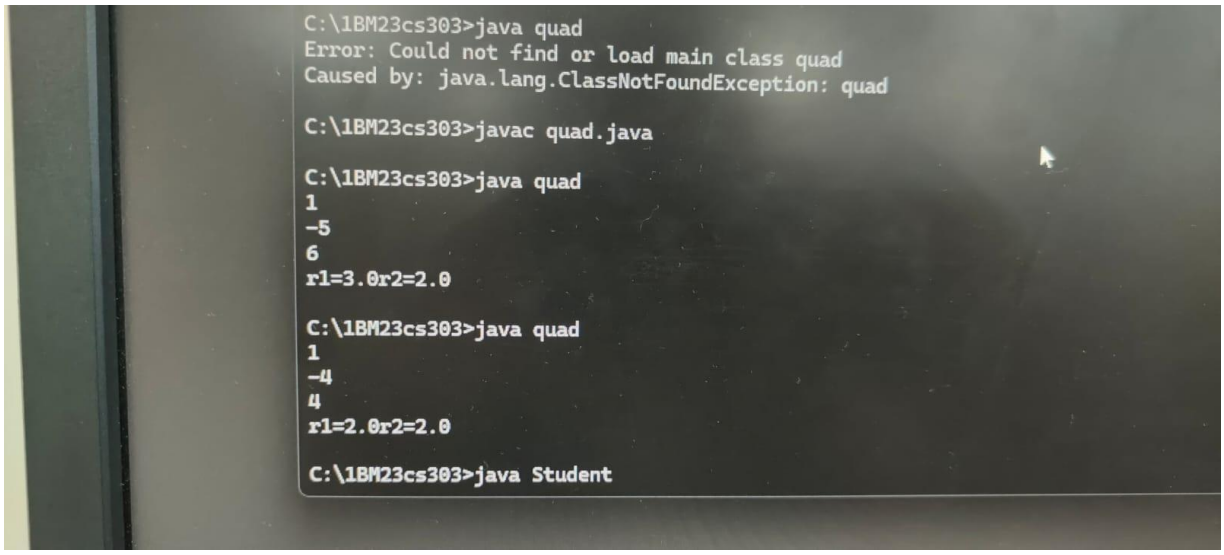
2

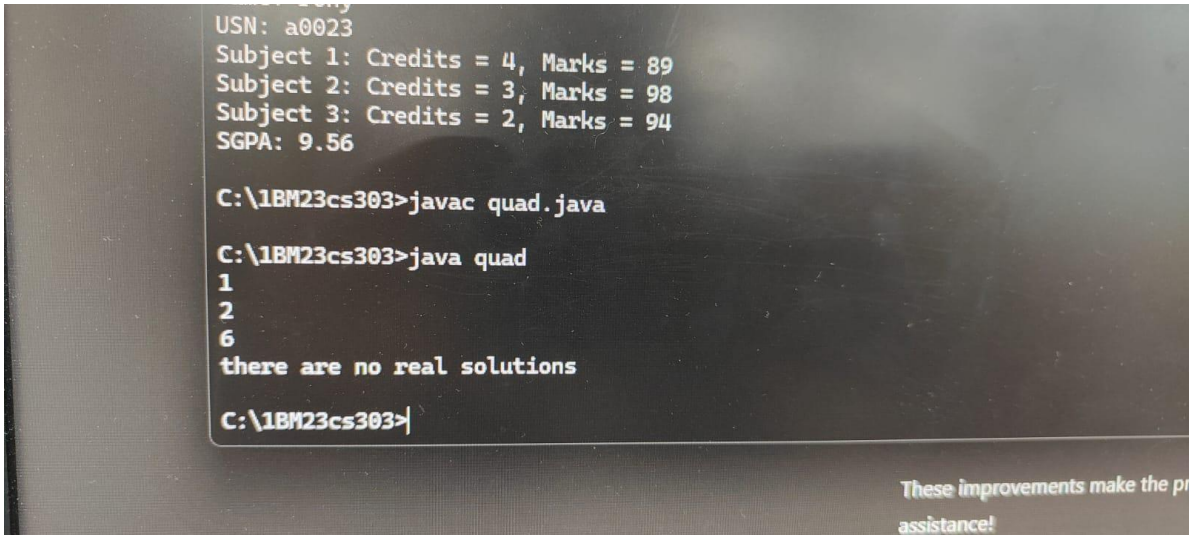
3

There are no real solutions.

26/9/24

```
import java.util.Scanner;
 class quad
{
  public static void main(String[] args)
{
    int d;
    int a,b,c;
    double r1,r2;
    Scanner ss=new Scanner(System.in);
    a=ss.nextInt();
    b=ss.nextInt();
    c=ss.nextInt();
    d=b*b-4*a*c;
    if(d>0)
    {
       r1=(-b+Math.pow(d,0.5))/(2*a);
       r2=(-b-Math.pow(d,0.5))/(2*a);
       System.out.println("r1="+r1+"r2="+r2);
    }
    else if(d==0)
    {
       r1=r2=-b/(2*a);
       System.out.println("r1="+r1+"r2="+r2);
    }
    else
    {
       System.out.println("there are no real solutions");
    }
}
}
```


C:\1BM23cs303>java quad
Error: Could not find or load main class quad
Caused by: java.lang.ClassNotFoundException: quad
C:\1BM23cs303>javac quad.java
C:\1BM23cs303>java quad
1
-5
6
r1=3.0r2=2.0
C:\1BM23cs303>java quad
1
-4
4
r1=2.0r2=2.0
C:\1BM23cs303>java Student


USN: a0023
Subject 1: Credits = 4, Marks = 89
Subject 2: Credits = 3, Marks = 98
Subject 3: Credits = 2, Marks = 94
SGPA: 9.56
C:\1BM23cs303>javac quad.java
C:\1BM23cs303>java quad
1
2
6
there are no real solutions
C:\1BM23cs303>
These improvements make the pr
assistance!

2) 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 {
    int cdt[], marks[];
    String sname, usn;
    Student() {
        cdt = new int[5];
        marks = new int[5];
    }
    void getd() {
    Scanner ss = new Scanner(System.in);
    sname = ss.next(); usn = ss.next();
    for (int i=0; i<3; i++)
    { System.out.println("Enter credits " + (i+1) + ":");
      cdt[i] = ss.nextInt();
      System.out.println("Enter marks " + (i+1) + ":");
      marks[i] = ss.nextInt();
    }
}
void display(){
 System.out.println("details are: \n Name:" + sname);
 System.out.println("USN:" + usn);
 for (int i=0; i<3; i++)
{ System.out.println("Credits:" + (i+1) + ":" + cdt[i]);
  System.out.println("Marks:" + (i+1) + ":" + marks[i]);
}
```

```
void calc()
{
int tc=0;
double tp=0.0;
for(int i=0;i<3;i++){
   tc+=cdt[i];
int gp=getpoint(marks[i]);
   tp+=gp*cdt[i];
}
double s=tp/tc;
System.out.println("sgpa:"+s);
}
int getpoints(int marks)
{
 if(marks>=90 && marks<=100)
    return 10;
 else if(marks>=80 && marks<=90)
    return 9;
 else if(marks>=70 && marks<=80)
    return 8;
  else if(marks>=60 && marks<=70)
    return 7;
  else if(marks>=50 && marks<=60)
    return 6;
  else if(marks>=40 && marks<=50)
    return 5;
  else
    return 0;
  }
}
```

```
public class main
{
public static void main
    (String[] args)
{
  Student s1=new Stude
  s1.getd();
  s1.display();
  s1.calc();
}
}
```

Output: Enter details:

Harry

1BM23CS106

Enter credits 1: 4

Enter marks 2: 89

Enter credits 2: 3

Enter marks 2: 93

Enter credits 3: 2

Enter marks 3: 94

Details are:

Name: Harry

USN: 1BM23CS106

Credits 1: 4

Marks 1: 89

Credits 2: 3

Marks 2: 93

Credits 3: 2

Credits : 94

Marks : 94

sgpa : 9.55

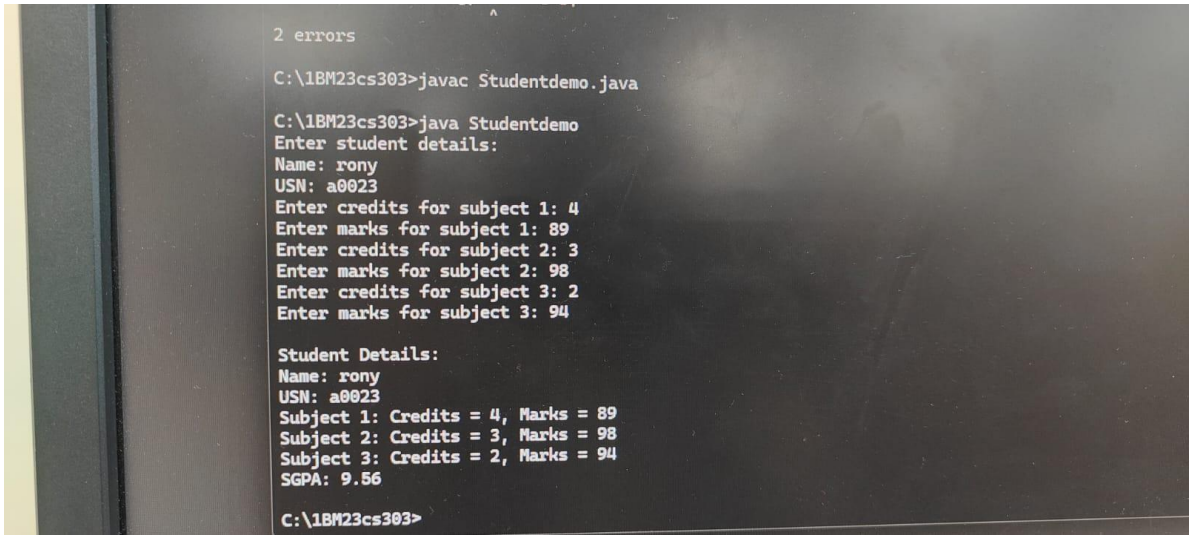
3/10/24

contains four

```
import java.util.Scanner;
class Student {

   int[] cdt, marks;
   String sname, usn;
   Student() {

      cdt = new int[10];

      marks = new int[10];

   }
   void getd() {

      Scanner ss = new Scanner(System.in);
      System.out.println("Enter student details:");
      System.out.print("Name: ");
      sname = ss.next();
      System.out.print("USN: ");
      usn = ss.next();
      for (int j = 0; j < 3; j++) {

         System.out.print("Enter credits for subject " + (j + 1) + ": ");
         cdt[j] = ss.nextInt();
         System.out.print("Enter marks for subject " + (j + 1) + ": ");
         marks[j] = ss.nextInt();

      }

   }
   void display() {

      System.out.println("\nStudent Details:");
      System.out.println("Name: " + sname);
      System.out.println("USN: " + usn);
      for (int j = 0; j < 3; j++) {

         System.out.println("Subject " + (j + 1) + ": Credits = " + cdt[j] + ", Marks = " + marks[j]);

      }

   }
void calc() {

      int tc = 0;
      double tp = 0.0;
      for (int i = 0; i < 3; i++) {
```

```
            tc += cdt[i];

            int gp = getpoints(marks[i]);

            tp+= gp*cdt[i];

        }
        double s = tp / tc;
        System.out.printf("SGPA: %.2f\n", s);

    }
    int getpoints(int marks) {

        if (marks >= 90 && marks <= 100)

            return 10;

        else if (marks >= 80 && marks < 90)

            return 9;

        else if (marks >= 70 && marks < 80)

            return 8;

        else if (marks >= 60 && marks < 70)

            return 7;

        else if (marks >= 50 && marks < 60)

            return 6;

        else if (marks >= 40 && marks < 50)

            return 5;

        else

            return 0;

    }

}
```

```
class Studentdemo {

    public static void main(String[] args) {

        Student s1 = new Student();
        s1.getd();
        s1.display();
        s1.calc();

    }
}
```



3) 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.

```
import java.util.Scanner;
class Book {
    String bname, author;
    int price, nop;
    Book() { }
    Book(String bnam, String auth, int p, int np)
{   bname = bnam;
    author = auth;
    price = p; nop = np;
}
    void setd()
{ Scanner ss = new Scanner(System.in);
  System.out.println("Enter book details:");
  bname = ss.next(); author = ss.next();
  price = ss.nextInt(); nop = ss.nextInt(); }
  void getd()
{
  System.out.println("Book details are:");
  System.out.println("\tName:" +bname);
  System.out.println("\tAuthor:" + author +
                     "\n price in nop);
}
  public String tostring()
{
  System.out.println("Details are:");
  return (bname + " " + author + " " + price + " :"
}
```

```
public class main
{
  public static void main (String [] args)
  {
    int n;
    Book b1 = new Book ("Harry Potter", "Rony",
                                    550, 235);
     b1.getd();
    Scanner sa = new Scanner (System.in);
    System.out.println ("Enter n:");
    n = sa.nextInt();
    Book b[] = new Book[n];
    for(int i=0; i<n; i++)
    {
       b[i] = new Book();
        b[i].setd();
    }
    for (int i=0; i<n; i++);
    {
      System.out.println(b[i].tostring());
    }
  }
}
```

Output:

```
Book details are: Harry Potter  Rony  550  235
Enter n: 3
Enter book details:
Ramayan
Ved
768
546
```

Enter book details:

Mahayud

Vyas

788

364

Enter book details:

Happy Soul

Michelle

678

381

Book details are:

Ramayan ved 768 546

Book details are:

Mahabbarat Vyas 788 234

Book details are:

HappySoul Michelle 678 381

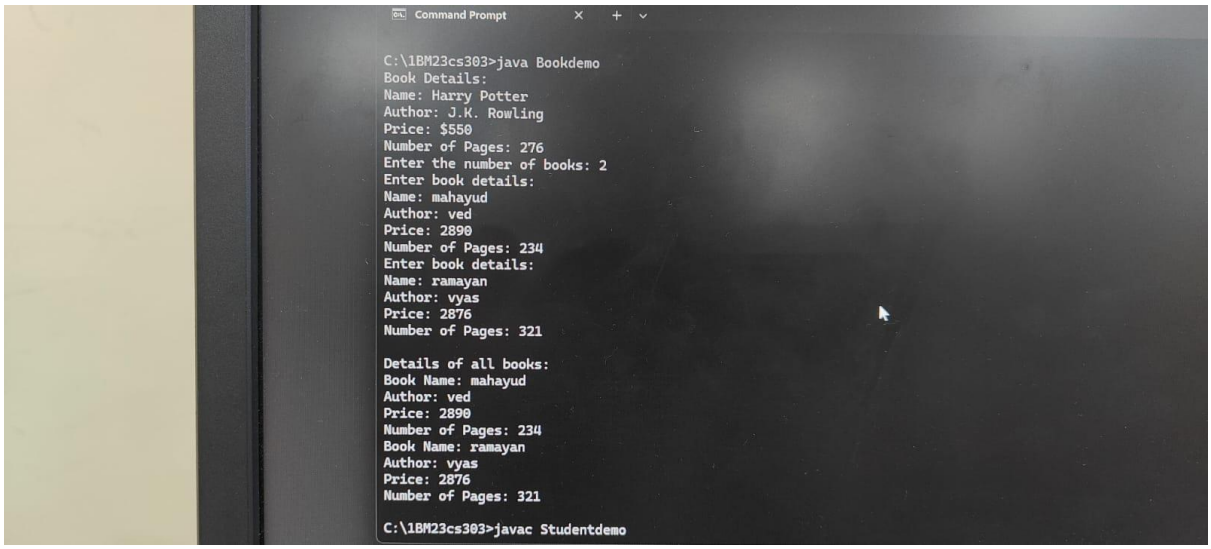
N

19/10/24

```
import java.util.Scanner;
class Book {

   String bname, author;
   int price, nop;
   Book() {}
   Book(String bname, String author, int price, int nop) {

      this.bname = bname;

      this.author = author;

      this.price = price;

      this.nop = nop;

   }

    void setd() {

      Scanner ss = new Scanner(System.in);

      System.out.println("Enter book details:");



      System.out.print("Name: ");

      bname = ss.next();



      System.out.print("Author: ");

      author = ss.next();



      System.out.print("Price: ");

      price = ss.nextInt();



      System.out.print("Number of Pages: ");

      nop = ss.nextInt();
```

```
    }

    void getd() {

        System.out.println("Book Details:");

        System.out.println("Name: " + bname);

        System.out.println("Author: " + author);

        System.out.println("Price: $" + price);

        System.out.println("Number of Pages: " + nop);

    }

     public String toString() {

        return "Book Name: " + bname + "\nAuthor: " + author + "\nPrice: " + price + "\nNumber of
Pages: " + nop;

    }

}

 class Bookdemo {

    public static void main(String[] args) {

        Book b1 = new Book("Harry Potter", "J.K. Rowling", 550, 276);

        b1.getd();

        Scanner sa = new Scanner(System.in);

        System.out.print("Enter the number of books: ");

        int n = sa.nextInt();


        Book[] books = new Book[n];

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

            books[i] = new Book();

            books[i].setd();
```

```
        }
        System.out.println("\nDetails of all books:");
        for (int i = 0; i < n; i++) {
            System.out.println(books[i].toString());
        }
    }
}
```



LAB PROGRAM - 4

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

```
import java.util.Scanner;
abstract class Shape
{
  int l,b; double a;
  Scanner ss=new Scanner(System.in);
   Shape()
  { System.out.println("enter l:");
    l = ss.nextInt();
     System.out.println("enter b:");
    b = ss.nextInt();

  }
  void printArea()
      {
      }
}

class Rectangle extends Shape
{
   void printArea()
   {
       a = l*b;
      System.out.println("Area:" + this.a);
   }
}
class Triangle extends Shape
{
  void printArea()
     {
       a = 0.5*l*b;
       System.out.println("Area:" + this.a);
     }
}
```

```
class Circle extends Shape
{
    void printArea()
    {
        area = 3.14 * l * l;
        System.out.println("area:" + this.a);
    }
}
class Shapedemo
{
    public static void main (String[] args)
    {
        Rectangle r = new Rectangle();
        Triangle t = new Triangle();
        Circle c = new Circle();
        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

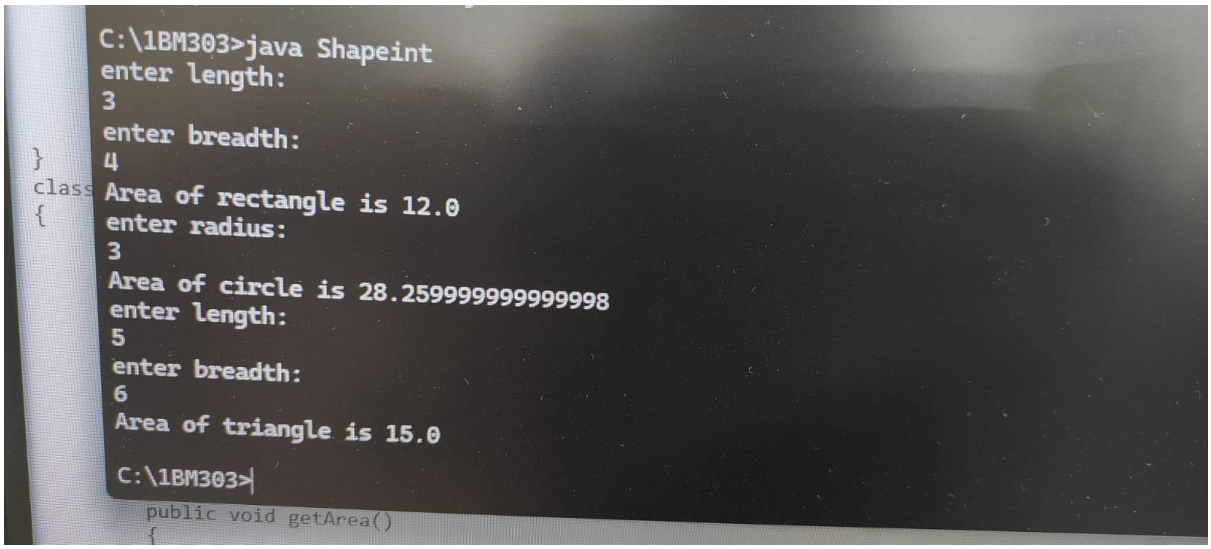
Output:

```
Enter l: 3
Enter b: 4
Enter l: 6
Enter b: 8
Enter l: 2
Enter b: 8
Area is : 12.0
area is : 24.0
area is: 12.56.
```

24/10/24

```
import java.util.Scanner;
abstract class Shape
{
   int l,b;
   double a;
   Scanner ss=new Scanner(System.in);
   Shape()
   {
   System.out.println("enter l:");
   l=ss.nextInt();
   System.out.println("enter b:");
   b=ss.nextInt();
   }
   void printArea()
   {

   }
}
class Rectangle extends Shape
{

   void printArea()
   {
      a=l*b;

      System.out.println("area is :"+this.a);
}
   }


class Triangle extends Shape
{
   void printArea()
   {
      a=0.5*l*b;
      System.out.println("area is :"+this.a);
   }

}
class Circle extends Shape
{
   void printArea()
   {
      a=3.14*l*l;
      System.out.println("area is :"+this.a);
   }

}
```

```
public class main
{
   public static void main(String[] args)
   {
    //Shape s1=new Shape();
    Rectangle s2=new Rectangle();
    Triangle s3=new Triangle();
    Circle s4=new Circle();
    //s1.printArea();
    s2.printArea();
    s3.printArea();
    s4.printArea();
   }
}
```



LAB PROGRAM - 5

5. balance and if the balance falls below this level, a service charge is imposed. Create a class Accoun that stores customer name, account number and type of account. From this derive the class Curacc and Savacct to make them more specific to the requirements. Include the below into methods:

a) Accept deposit from customer and update bal
b) Display the balance.
c) Compute and deposit interest.
d) Permit the withdrawal and update the balance.

```
import java.util.Scanner;
class Amount
{
  String custname;
  String accno; double deposit; double bal;
  double wamt;
  void getd()
  {
    Scanner s = new Scanner(System.in);
  System.out.println("Enter the customer name:");
  custname = sc.nextline();
  System.out.println("Enter the customer account
                                    number:");
  accno = sc.nextline();
  System.out.println("Enter deposit amount:");
  deposit = sc.nextDouble();
  bala = deposit;
  System.out.println();
  }
  void putd()
  {
  System.out.println("Customer name:" + custname);
  System.out.println("Account no:" + accno);
  }}
```

```
class Curacct extends Account {
    void balancecheck()
    {
        if (balance <= 10000)
        { System.out.println("You have less
            than minimum balance");
          balance -= 10000;
        }
    }
void calcbal() {
 System.out.println("current account details:");
  putd();
 System.out.println("Enter amount to withdraw");
  Scanner ss = new Scanner(System.in);
  wamt = ss.nextDouble();
  bal -= wamt;
  balancecheck();
  System.out.println("Balance after penalty" + bal);
  System.out.println();
 }
}
class Savacct extends Account {
    void calcint(){
     bal = bal + (0.07 * bal);
    }
void calcbal(){
 Scanner sc = new Scanner(System.in);
 System.out.println("Enter amount to withdraw");
                    Savings account details
 putd();
 System.out.println("Enter amount to withdraw:");
 wamt = sc.nextDouble();
 bal -= wamt;
```

```
System.out.println(" Balance before compound interest:"
                                                + bal);
calcint();
System.out.println("Balance after compound interest:"
                                                + bal);
}
}
class Bank {
    public static void main(String args[])
    {
      Scanner ss = new Scanner(System.in);
      int acctype;
    System.out.println(" enter 1 for savings account
      and 2 for current account");
      acctype = ss.nextInt();
      if (acctype==1) {
      Savacct sa = new Savacct();
      sa.getd();
      sa.calcbal();
    }
    else if (acctype==2) {
        cuacct ca = new Cuacct();
        ca.getd();
        ca.calcbal();
        System.out.println("Check book is provided");
    }
    else {
      System.out.println("enter a valid account
                                        type");
      }
    }
}
```

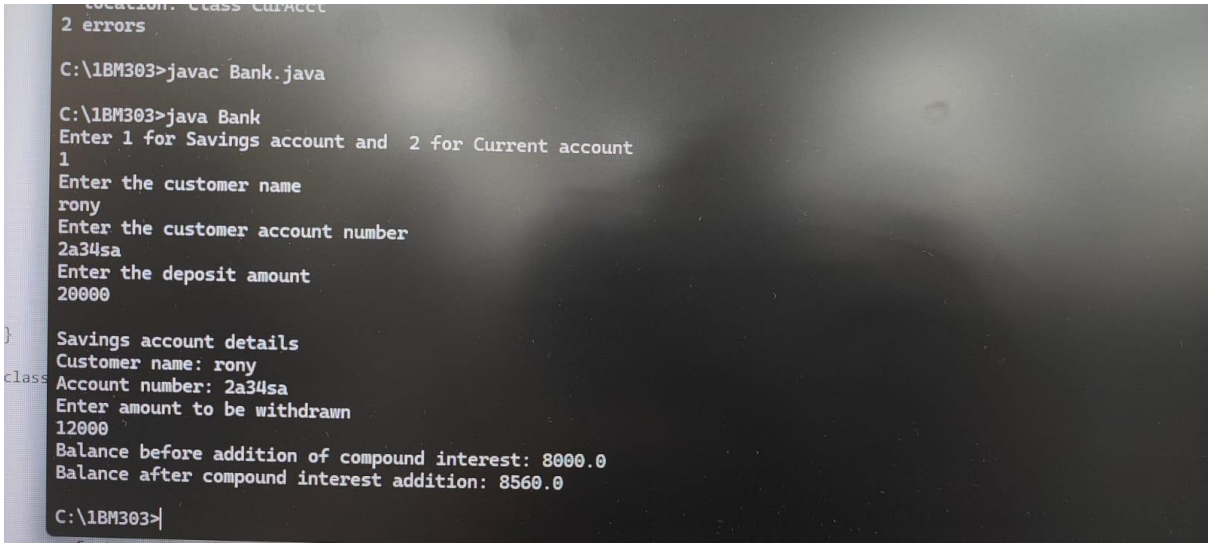
Output: Enter 1 for savings account and 2 for current account: 1
Enter the customer name: Rony
Enter the customer account number:
2A43BM36
Enter the deposit amount
20000
Savings account details:
Customer name: Rony
Account number: 2A43BM36
Enter amount to be withdrawn
12000
Balance before addition of compound interest: 80000
Balance after compound interest addition: 8560.0

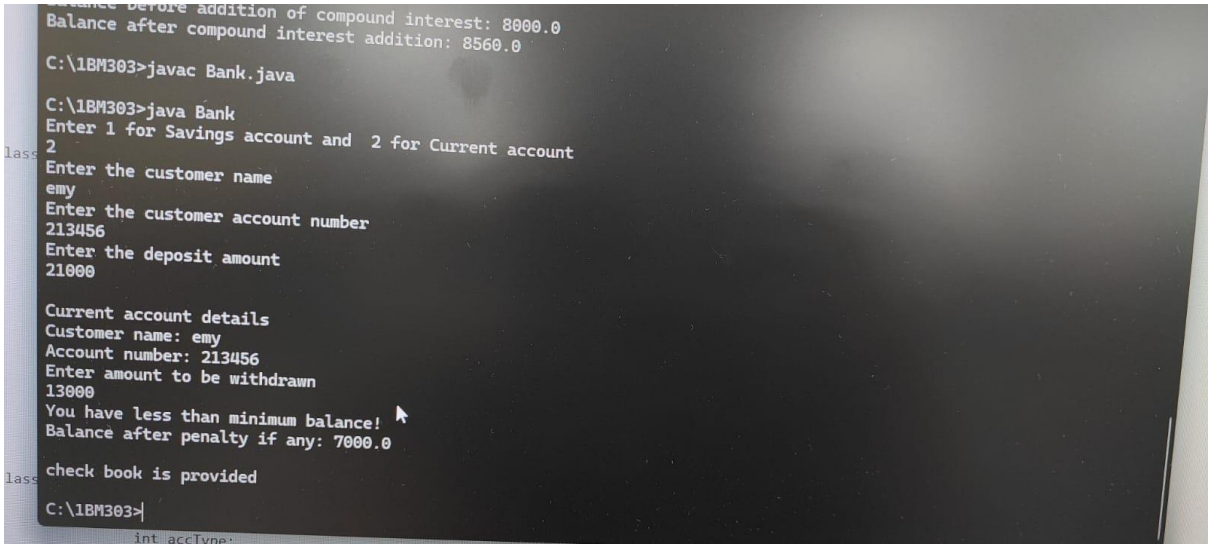
Enter 1 for savings account and 2 for current account:
2
Enter the customer name: Emy
Enter the customer account number:
2A43BM72
Enter the deposit amount:
21000
Current account details:
Customer name: Emy
Account number: 2A43BM72
Enter amount to be withdrawn
13000
You have less than minimum balance!
Balance after penalty: 7000.0

21/11/24

```
import java.util.Scanner;
class Account {
String custName;
String accno;
double deposit;
double bal;
double wamt;

void getd() {
Scanner sc = new Scanner(System.in);
System.out.println("Enter the customer name");
custName = sc.nextLine();
System.out.println("Enter the customer account number");
accno = sc.nextLine();
System.out.println("Enter the deposit amount");
deposit = sc.nextDouble();
bal=deposit;
System.out.println();

}
void putd() {
System.out.println("Customer name: "+custName);
System.out.println("Account number: "+accno);
//System.out.println();
}
}

class CurAcct extends Account {
void balanceCheck() {
if (bal<=10000) {
System.out.println("You have less than minimum balance!");
bal-=1000;
}
}
void calcDisplayBalance() {
System.out.println("Current account details");
putd();
System.out.println("Enter amount to be withdrawn");
Scanner sc = new Scanner(System.in);
wamt = sc.nextDouble();
bal-=wamt;
balanceCheck();
System.out.println("Balance after penalty if any: "+bal);
System.out.println();
}
}

class SavAcct extends Account {
```

```
void interestCalc() {
bal=bal+(0.07*bal);
}
void calcDisplayBalance() {
Scanner sc = new Scanner(System.in);
System.out.println("Savings account details");
putd();
System.out.println("Enter amount to be withdrawn");
wamt = sc.nextDouble();
bal-=wamt;
System.out.println("Balance before addition of compound interest: "+bal);
interestCalc();
System.out.println("Balance after compound interest addition: "+bal);
}
}

class Bank {
public static void main(String args[]) {
Scanner sc = new Scanner(System.in);
int accType;
System.out.println("Enter 1 for Savings account and  2 for Current account");
accType = sc.nextInt();
if (accType==1) {
SavAcct sacc = new SavAcct();
sacc.getd();
sacc.calcDisplayBalance();

}
else if (accType==2) {
CurAcct cacc = new CurAcct();
cacc.getd();
cacc.calcDisplayBalance();
System.out.println("check book is provided ");
}
else {
System.out.println("Enter a valid account type");
}
}
}
```


2 errors
C:\1BM303>javac Bank.java
C:\1BM303>java Bank
Enter 1 for Savings account and 2 for Current account
1
Enter the customer name
rony
Enter the customer account number
2a34sa
Enter the deposit amount
20000
Savings account details
Customer name: rony
Account number: 2a34sa
Enter amount to be withdrawn
12000
Balance before addition of compound interest: 8000.0
Balance after compound interest addition: 8560.0
C:\1BM303>


Balance before addition of compound interest: 8000.0
Balance after compound interest addition: 8560.0
C:\1BM303>javac Bank.java
C:\1BM303>java Bank
Enter 1 for Savings account and 2 for Current account
2
Enter the customer name
emy
Enter the customer account number
213456
Enter the deposit amount
21000
Current account details
Customer name: emy
Account number: 213456
Enter amount to be withdrawn
13000
You have less than minimum balance!
Balance after penalty if any: 7000.0
check book is provided
C:\1BM303>

LAB PROGRAM - 6

6. Create a package CIE which has two classes- student and Internals. The class student 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.

File ①

```
package cie;
import java.util.Scanner;
public class Student
{
    public int usn, sem;
    public String sname;
    Scanner ss = new Scanner(System.in);
    public void getdc()
    {
     System.out.println("Enter name:");
     sname = ss.next();
     System.out.println("Enter usn:");
      usn = ss.nextInt();
     System.out.println("Enter sem:");
      sem = ss.nextInt();
    }
    public void disp()
    {
      System.out.println("name:"+sname);
      System.out.println("usn:" + usn);
    } } System.out.println("Sem:" + sem);
```

file ②

```
package cie;
import java.util.Scanner;
public class Internals
{
    public int marks[] = new int[5];
    Scanner sa = new Scanner(System.in);
    public void getdl();
    {
        System.out.println("Enter internal marks:");
        for(int i=0; i<5; i++)
        {
            System.out.println("Enter marks of course"
                               + (i+1) + ":");
            marks[i] = sa.nextInt();
        }
    }
public void displ()
{
    System.out.println("Internals marks");
    for(int i=0; i<5; i++)
    {
        System.out.println("Marks of course" + (i+1) +
                           + marks[i]);
    }
}
}
```

file ③.

```
package see;
import cie.Student;
import java.util.Scanner;
public class External extends Student
{
    public int m[] = new int[5];
    Scanner sr = new Scanner(System.in);
```

```
public void getd2()
{
    System.out.println("Enter see marks:");
    for(int j=0;j<5;j++)
    {
        System.out.println("Enter marks of course"+
                           (j+1)+":");
        m[j]=ss.nextInt();
    }
}
public void disp2()
{
  System.out.println("See Marks:");
  for (int j=0; j<5; j++)
  {
   System.out.println("Marks of course "+(j+1)+":"+
                                        m[j]);
  }
}
}
```

file ④.

```
import cie.Internals;
import cie.Student;
import see.External;
import java.util.Scanner;
class Finals
{
  public static void main(String args[])
  {
   int n;
   Scanner se=new Scanner(System.in);
   int mark[]=new int[5];
   Student s=new Student();
   Internals it=new Internals();
   External et=new External();
   System.out.println("Enter n:");
    n=se.nextInt();
```

```
for (int i=0; i<n; i++)
{
    s.getd();    it.getd1();    et.getd2();
    s.disp();    it.disp1();    et.disp2();
    System.out.println("Marks of " + s.sname + " is");
    for (int k=0; k<5; k++)
    {
        mark[k] = (it.marks[k]) + (et.m[k]);
        System.out.println("final marks in course"
        + (k+1) + ":" + mark[k]);
    }
}
}
```

Output:

```
Enter n:
2
Enter usn:
345
Enter sem:
3
Enter internals marks:
Enter marks of course 1:
44
Enter marks of course 2:
45
Enter marks of course 3:
46
Enter marks of course 4:
47
Enter marks of course 5:
48
Enter see marks:
Enter marks of course 1:
49
Enter marks of course 2:
48
Enter marks of course 3:
47
Enter marks of course 4:
45
Enter marks of course 5:
46
Name: Rony
usn: 345
sem: 3
Internals marks:
Marks of course 1: 44
Marks of course 2: 45
Marks of course 3: 46
Marks of course 4: 47
Marks of course 5: 48
See Marks:
Marks of course 1: 49
Marks of course 2: 48
Marks of course 3: 47
Marks of course 4: 45
Marks of course 5: 46
Marks of Rony "is
Final marks in course 1: 93
Final marks in course 2: 93
Final marks in course 3: 93
Final marks in course 4: 92
Final marks in course 5: 94
```

```
Enter name:
Emy
Enter usn:
234
Enter sem:
4
Enter internals marks:
Enter marks of course 1:
45
Enter marks of course 2:
46
Enter marks of course 3:
49
Enter marks of course 4:
48
Enter marks of course 5:
47
Enter see marks:
Enter marks of course 1:
50
Enter marks of course 2:
47
Enter marks of course 3:
48
Enter marks of course 4:
46
Enter marks of course 5:
45
Name : Lily
usn: 234
Sem: 4
Internals Marks:
Marks of course 1:45
Marks of course 2:46
Marks of course 3:49
Marks of course 4:48
Marks of course 5:47
See Marks:
Marks of course 1:50
Marks of course 2: 47
Marks of course 3: 48
Marks of course 4:46
Marks of course 5: 45
Marks of Lily is:
Final marks in course 1:95
Final marks in course 2:93
Final marks in course 3:97
Final marks in course 4:94
Final marks in course 5:92
```

21/11/24

```
package cie;
import java.util.Scanner;
public class Student
{
public int usn,sem;
public String sname;
Scanner ss=new Scanner(System.in);
public void getd()
{
System.out.println("enter name:");
sname=ss.next();
System.out.println("enter usn:");
usn=ss.nextInt();
System.out.println("enter sem:");
sem=ss.nextInt();
}
public void disp()
{

System.out.println("name:"+sname);
System.out.println("usn:"+usn);
System.out.println("sem:"+sem);
}
}
```

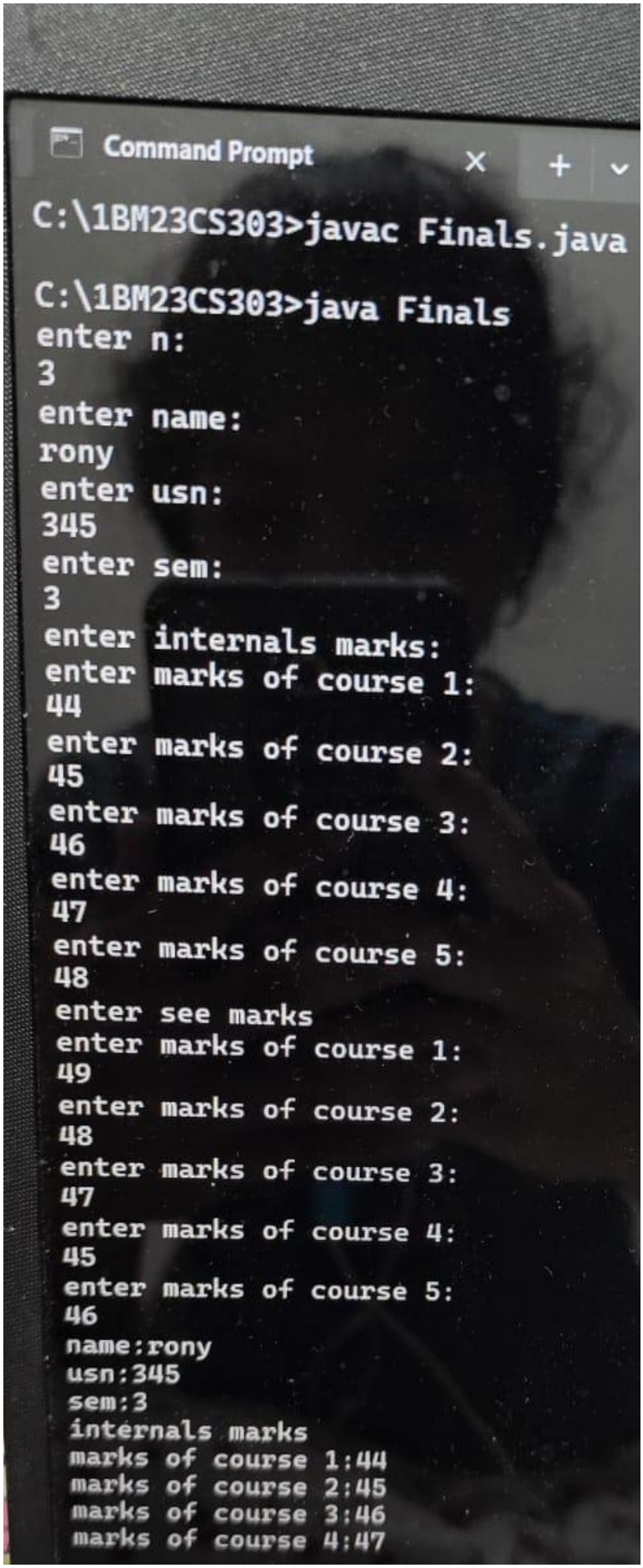
```
package cie;
import java.util.Scanner;
public class Internals
{
public int marks[]=new int[5];
Scanner sa=new Scanner(System.in);

public void getd1()
{
System.out.println("enter internals marks:");
for(int i=0;i<5;i++)
{
System.out.println("enter marks of course "+(i+1)+":");
marks[i]=sa.nextInt();
}
}
public void disp1()
{
System.out.println("internals marks");
for(int i=0;i<5;i++)
{
System.out.println("marks of course "+(i+1)+":"+marks[i]);
}
```

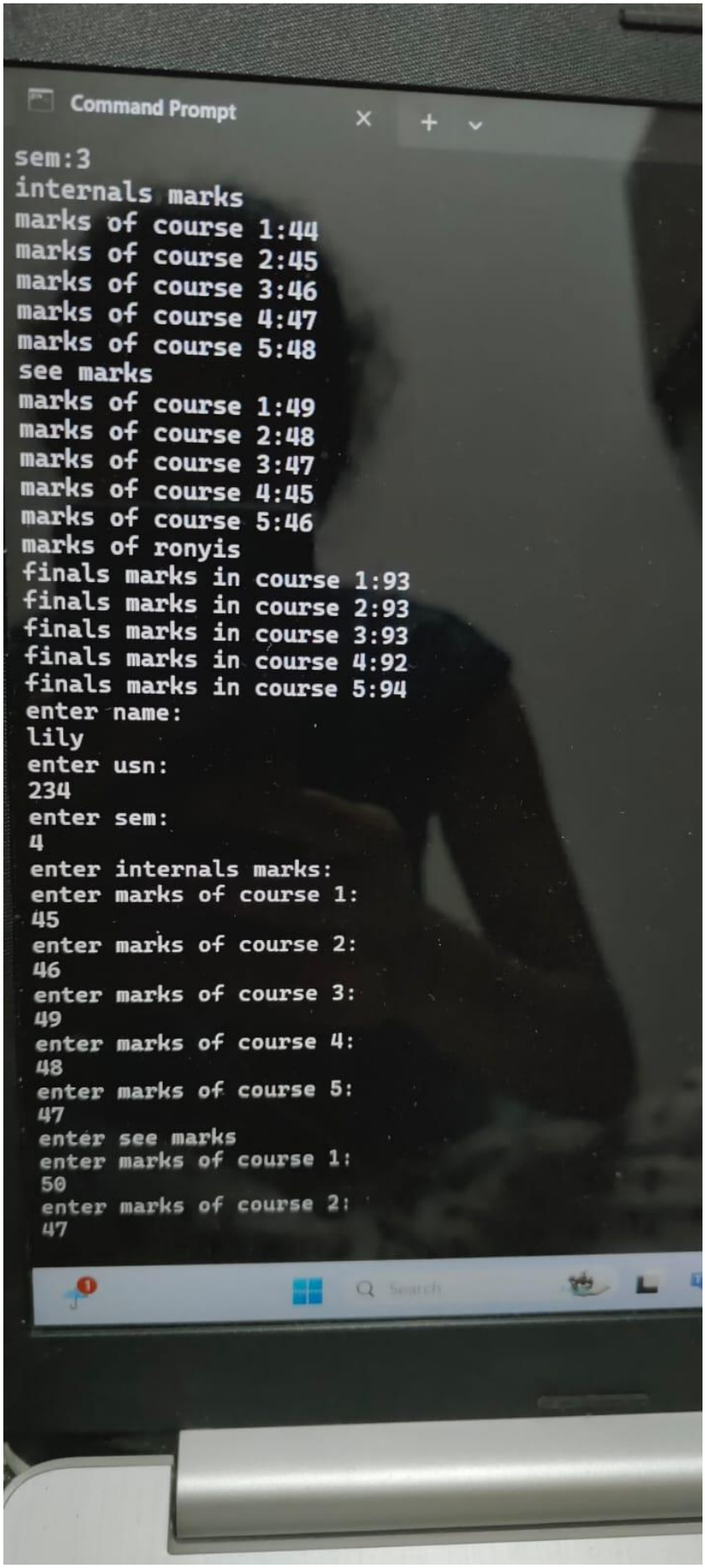
```
}
}
```

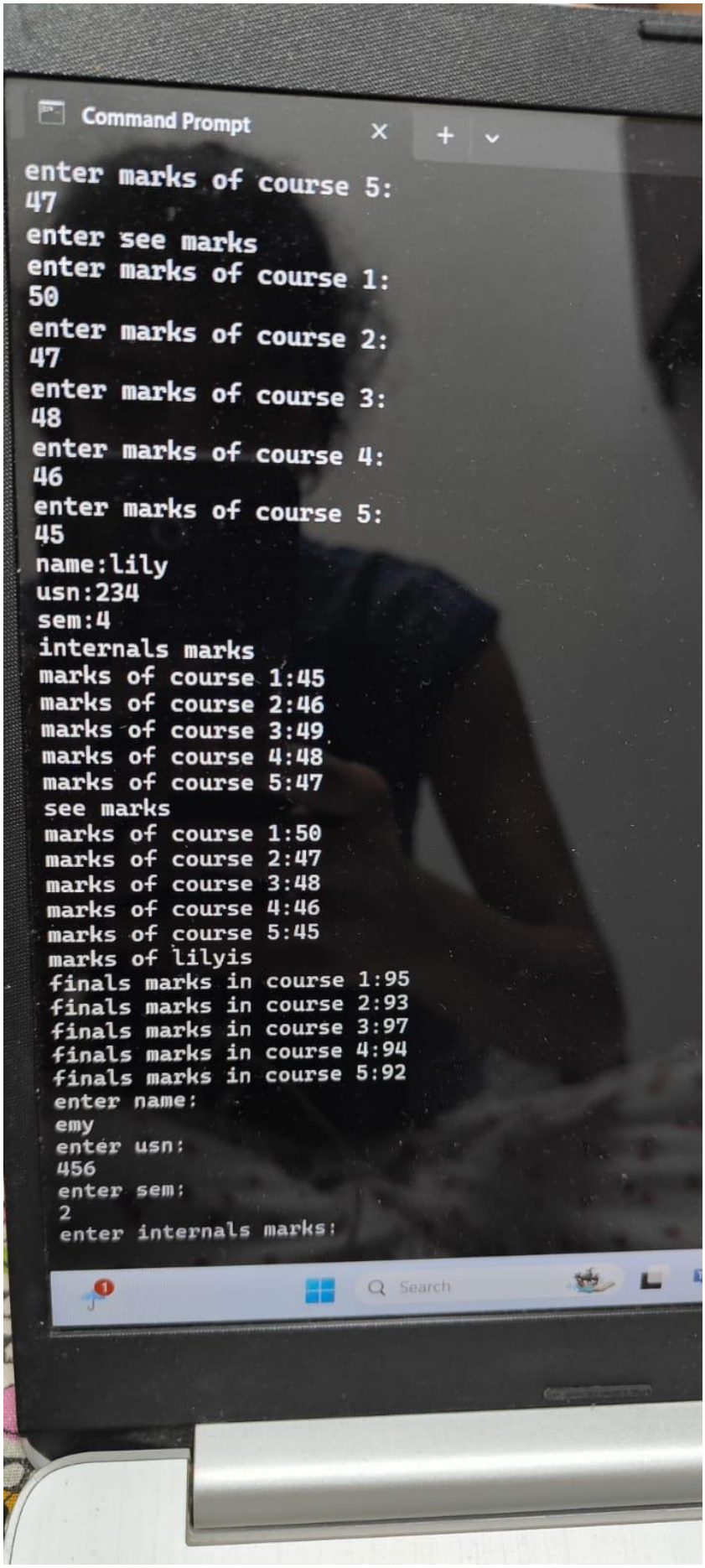
```
package see;
import cie.Student;
import java.util.Scanner;
public class External extends Student
{
public int m[]=new int[5];
Scanner sr=new Scanner(System.in);

public void getd2()
{
System.out.println("enter see marks");
for(int j=0;j<5;j++)
{
System.out.println("enter marks of course "+(j+1)+":");
m[j]=sr.nextInt();
}
}
public void disp2()
{
System.out.println("see marks");

for(int j=0;j<5;j++)
{
System.out.println("marks of course "+(j+1)+":"+m[j]);
}
}
}
```

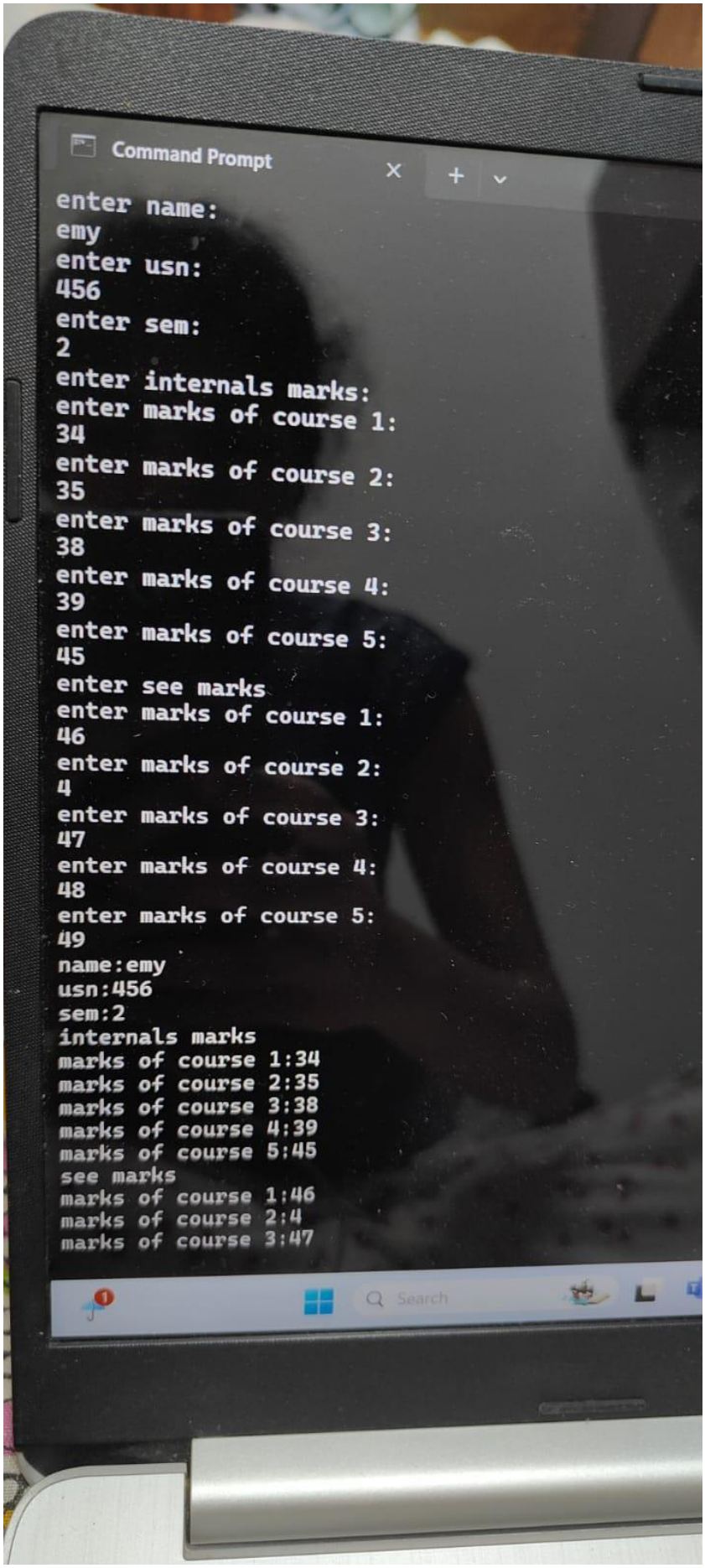
```
import cie.Internals;
import cie.Student;
import see.External;
import java.util.Scanner;
class Finals
{
public static void main(String args[])
{
int n;
Scanner se=new Scanner(System.in);
int mark[]=new int[5];
Student s=new Student();
Internals it=new Internals();
External et=new External();

System.out.println("enter n:");
n=se.nextInt();
for(int f=0;f<n;f++)
```

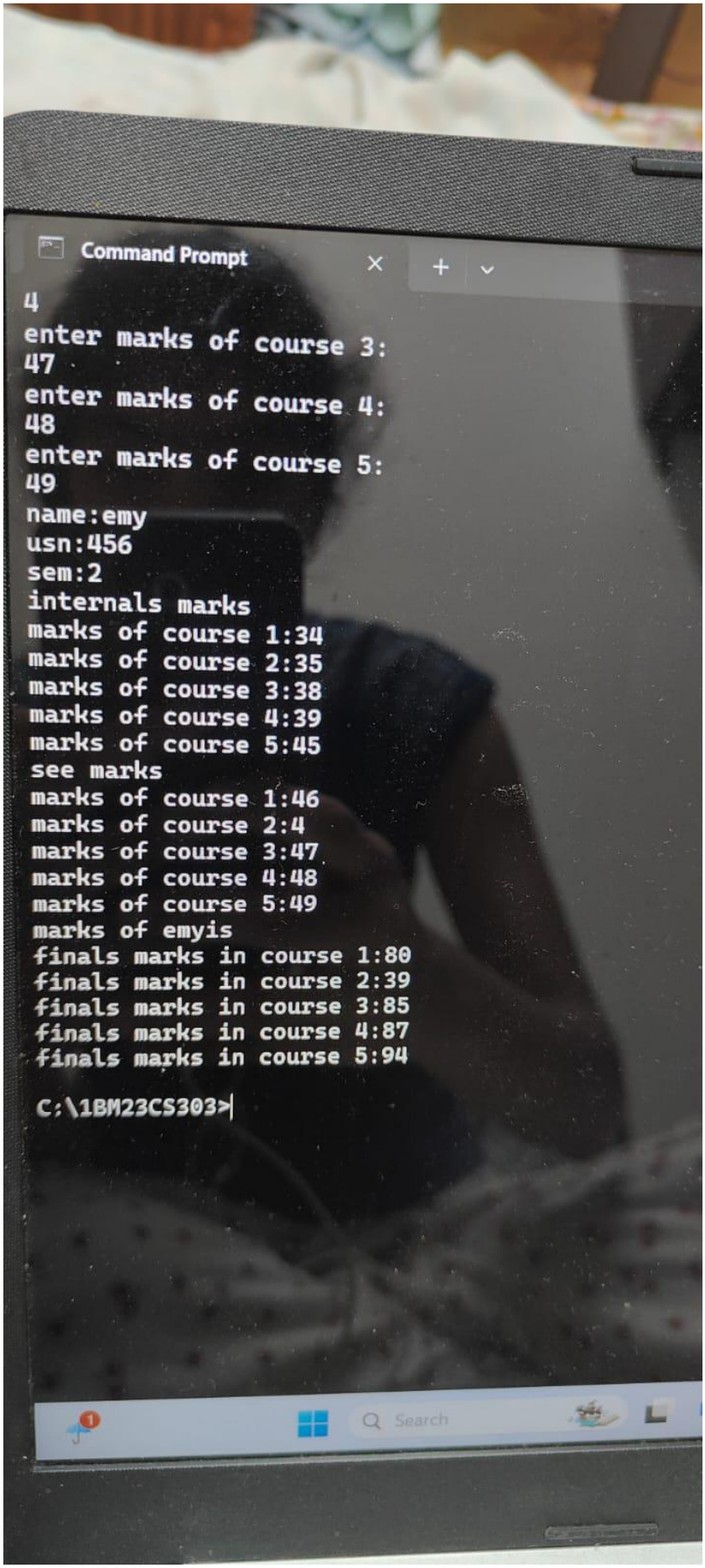
```
{

s.getd();
it.getd1();
et.getd2();
s.disp();
it.disp1();
et.disp2();
System.out.println("marks of "+s.sname+"is");
for(int k=0;k<5;k++)
{
mark[k]=(it.marks[k])+(et.m[k]);
System.out.println("finals marks in course "+(k+1)+":"+mark[k]);
}
}
}
}
```


Command Prompt
C:\1BM23CS303>javac Finals.java
C:\1BM23CS303>java Finals
enter n:
3
enter name:
rony
enter usn:
345
enter sem:
3
enter internals marks:
enter marks of course 1:
44
enter marks of course 2:
45
enter marks of course 3:
46
enter marks of course 4:
47
enter marks of course 5:
48
enter see marks
enter marks of course 1:
49
enter marks of course 2:
48
enter marks of course 3:
47
enter marks of course 4:
45
enter marks of course 5:
46
name:rony
usn:345
sem:3
internals marks
marks of course 1:44
marks of course 2:45
marks of course 3:46
marks of course 4:47


Command Prompt
sem:3
internals marks
marks of course 1:44
marks of course 2:45
marks of course 3:46
marks of course 4:47
marks of course 5:48
see marks
marks of course 1:49
marks of course 2:48
marks of course 3:47
marks of course 4:45
marks of course 5:46
marks of ronyis
finals marks in course 1:93
finals marks in course 2:93
finals marks in course 3:93
finals marks in course 4:92
finals marks in course 5:94
enter name:
lily
enter usn:
234
enter sem:
4
enter internals marks:
enter marks of course 1:
45
enter marks of course 2:
46
enter marks of course 3:
49
enter marks of course 4:
48
enter marks of course 5:
47
enter see marks
enter marks of course 1:
50
enter marks of course 2:
47


Command Prompt
enter marks of course 5:
47
enter see marks
enter marks of course 1:
50
enter marks of course 2:
47
enter marks of course 3:
48
enter marks of course 4:
46
enter marks of course 5:
45
name:lily
usn:234
sem:4
internals marks
marks of course 1:45
marks of course 2:46
marks of course 3:49
marks of course 4:48
marks of course 5:47
see marks
marks of course 1:50
marks of course 2:47
marks of course 3:48
marks of course 4:46
marks of course 5:45
marks of lilyis
finals marks in course 1:95
finals marks in course 2:93
finals marks in course 3:97
finals marks in course 4:94
finals marks in course 5:92
enter name:
emy
enter usn:
456
enter sem:
2
enter internals marks:
Search


Command Prompt
enter name:
emy
enter usn:
456
enter sem:
2
enter internals marks:
enter marks of course 1:
34
enter marks of course 2:
35
enter marks of course 3:
38
enter marks of course 4:
39
enter marks of course 5:
45
enter see marks
enter marks of course 1:
46
enter marks of course 2:
4
enter marks of course 3:
47
enter marks of course 4:
48
enter marks of course 5:
49
name:emy
usn:456
sem:2
internals marks
marks of course 1:34
marks of course 2:35
marks of course 3:38
marks of course 4:39
marks of course 5:45
see marks
marks of course 1:46
marks of course 2:4
marks of course 3:47
Search


Command Prompt
4
enter marks of course 3:
47
enter marks of course 4:
48
enter marks of course 5:
49
name:emy
usn:456
sem:2
internals marks
marks of course 1:34
marks of course 2:35
marks of course 3:38
marks of course 4:39
marks of course 5:45
see marks
marks of course 1:46
marks of course 2:4
marks of course 3:47
marks of course 4:48
marks of course 5:49
marks of emyis
finals marks in course 1:80
finals marks in course 2:39
finals marks in course 3:85
finals marks in course 4:87
finals marks in course 5:94
C:\1BM23CS303>
Search

LAB PROGRAM - 7

7. 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 uses both father and son's age and throws an exception if son's age is >= father's age.

```
import java.util.Scanner;
class WrongAgeException extends Exception
{
    WrongAgeException()
    {
        System.out.println("Age cannot be negative");
    }
}
class WrongSonAgeException extends Exception
{
    WrongSonAgeException()
    {
        System.out.println("son's age cannot be great
                            than father");
    }
}
class Father
{
    int fage;
    Scanner ss = new Scanner(System.in);
    Father() throws WrongAgeException
    {
        System.out.println("enter father's age");
        fage = ss.nextInt();
```

```
        if (fage < 0)
        {
            throw new WrongAgeException();
        }
        System.out.println("Father's age is" + fage);
    }
}
class Son extends Father
{
    int sage;
    Scanner ss = new Scanner (System.in);
    Son() throws WrongSonAgeException, WrongAgeExc
    {
        super();
        System.out.println("Enter son's age:");
        sage = ss.nextInt();
        if (sage < 0)
        {
            throw new WrongAgeException();
        }
        if (sage >= fage)
        {
            throw new WrongSonAgeException();
        }
        System.out.println("Son's age is" + sage);
    }
}
class Pgmdemo
{
    public static void main(String args[])
    {
        try
        {
            Son s = new Son();
        }
```

```
catch(Exception e)
{
    System.out.println("ages are not given correct");
}
}
}
```

Output 1: Enter father's age:
45
Father's age is 45
Enter son's age:
23
Son's age is 23

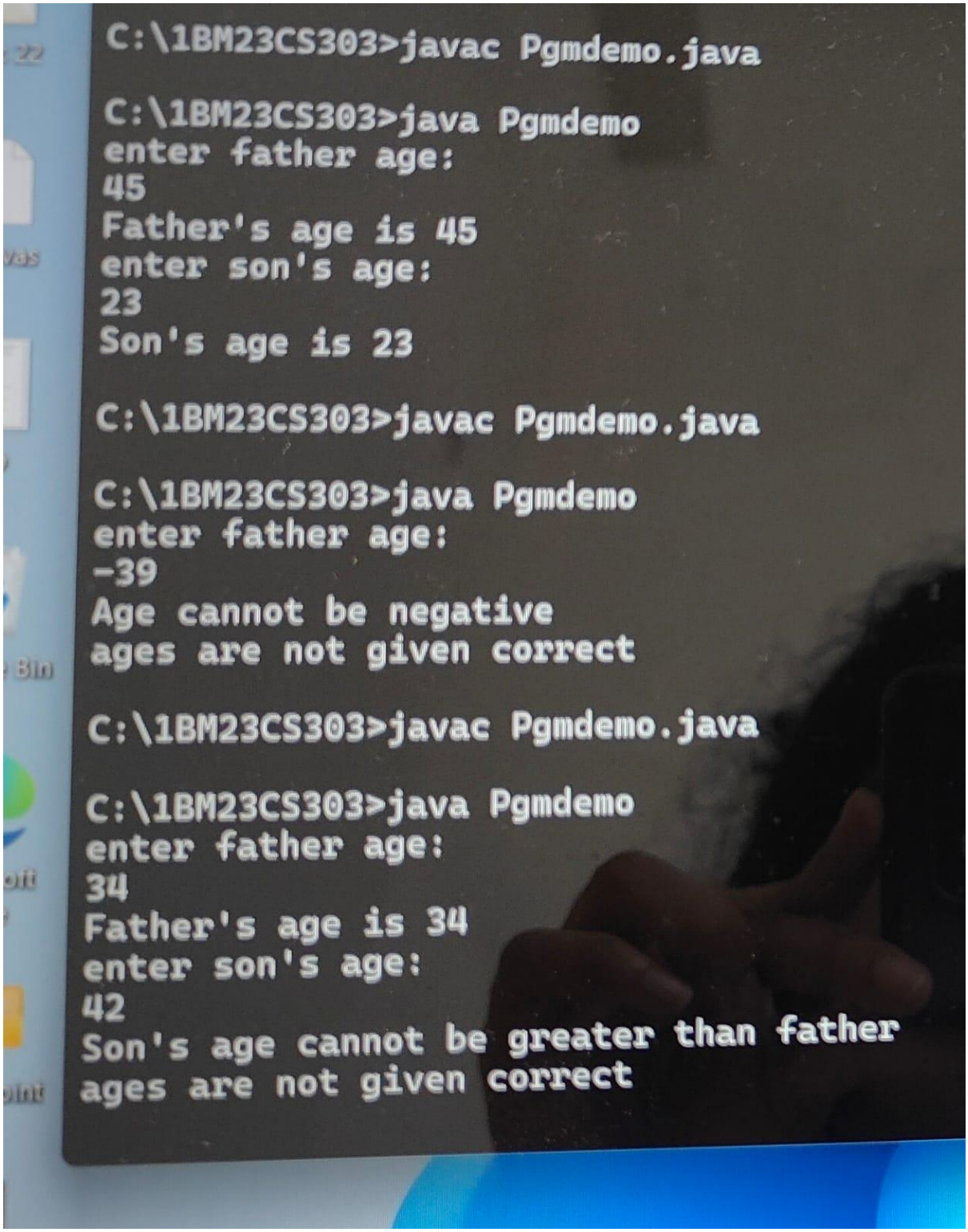
Output 2: Enter father's age:
-39
Age cannot be negative.
Ages are not given correct.

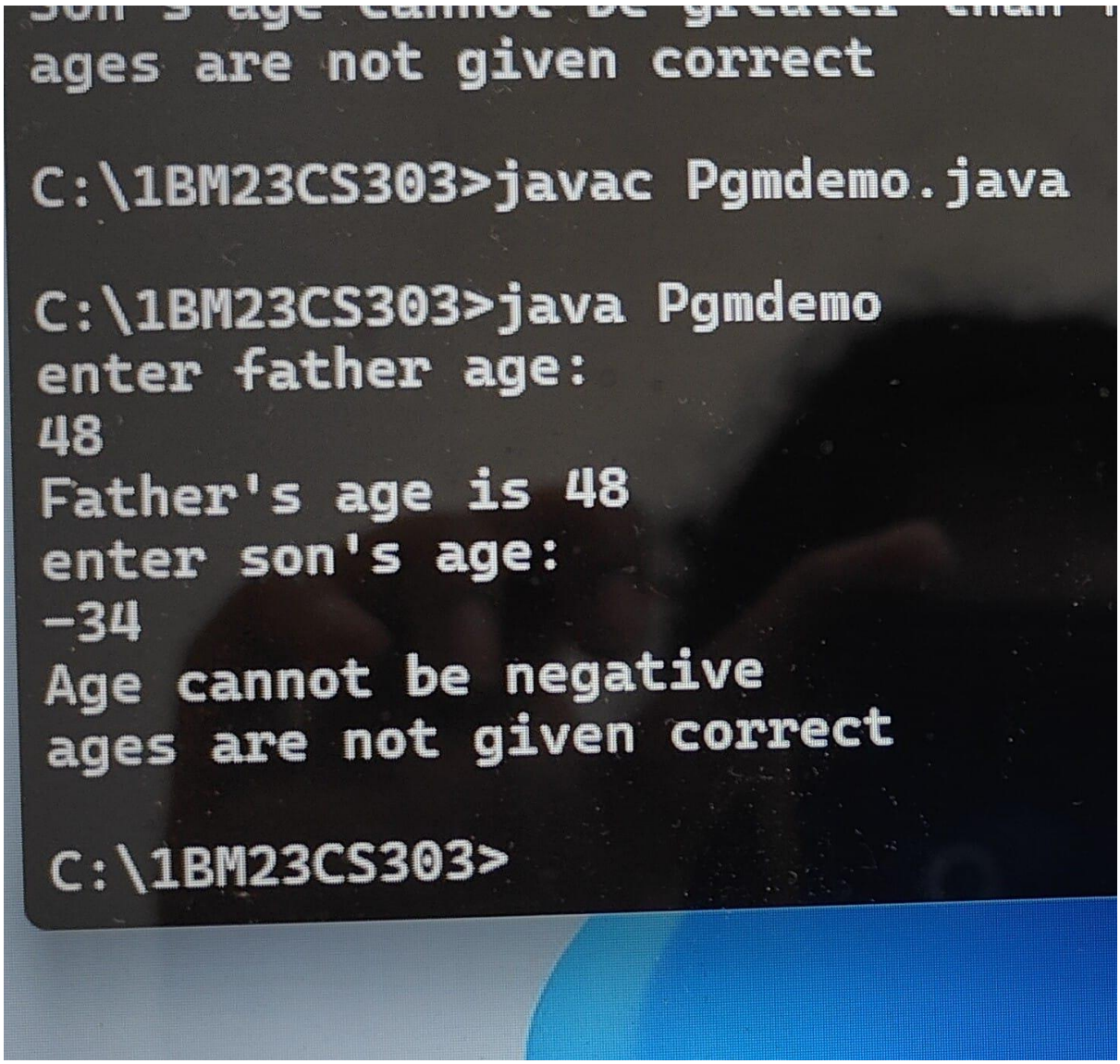
Output: 3: Enter father's age:
34
Father's age is 34
Enter son's age:
42
Son's age cannot be greater than fathe
Ages are not given correct.

21/11/24

```
import java.util.Scanner;
class WrongIntException extends Exception
{
WrongIntException()
{
System.out.println("Internal marks exceeded");
}
}
class WrongExtException extends Exception
{
WrongExtException()
{
System.out.println("External marks exceeded");
}
}
class Internals
{
int im;
Scanner ss=new Scanner(System.in);
Internals() throws WrongIntException
{
System.out.println("enter internal marks:");
im=ss.nextInt();
if (im>30)
{
throw new WrongIntException();
}
System.out.println("Internal marks are "+im);
}
}
class Externals
{
int em;
Scanner sa=new Scanner(System.in);
Externals() throws WrongExtException
{
System.out.println("enter external marks:");
em=sa.nextInt();
if (em>70)
{
throw new WrongExtException();
}
System.out.println("External marks are "+em);
}
}
class Marks
{
public static void main(String args[])
```

```
{
try
{
Internals i=new Internals();
Externals e=new Externals();
}
catch(Exception e)
{
System.out.println("marks entered are wrong");
}
}
}
```


C:\1BM23CS303>javac Pgmdemo.java
C:\1BM23CS303>java Pgmdemo
enter father age:
45
Father's age is 45
enter son's age:
23
Son's age is 23
C:\1BM23CS303>javac Pgmdemo.java
C:\1BM23CS303>java Pgmdemo
enter father age:
-39
Age cannot be negative
ages are not given correct
C:\1BM23CS303>javac Pgmdemo.java
C:\1BM23CS303>java Pgmdemo
enter father age:
34
Father's age is 34
enter son's age:
42
Son's age cannot be greater than father
ages are not given correct


ages are not given correct
C:\1BM23CS303>javac Pgmdemo.java
C:\1BM23CS303>java Pgmdemo
enter father age:
48
Father's age is 48
enter son's age:
-34
Age cannot be negative
ages are not given correct
C:\1BM23CS303>

LAB PROGRAM - 8

8. 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.

```
import java.util.Scanner;
class Thread1 extends Thread
{
  public void run()
  {
    try {
    System.out.println("BMS College of Engineering");
    Thread.sleep(10000);
    }
    catch(InterruptedException ie)
    {
      System.out.println("exit thread");
    }
  }
}
class Thread2 extends Thread
{
  public void run()
  {
    try {
     System.out.println("CSE");
     Thread.sleep(2000);
  }
  catch(InterruptedExecution ie)
  {
    System.out.println("Exit thread");
  }
}
}
```

```
class Tdpgm
{
  public static void main(String xx[])
  {
    Thread1 t1 = new Thread1();
    t1.start();
    Thread t2 = new Thread2();
     t2.start();
  }
}
```

Output: BMS College of Engineering
CSE.

Using only one class:

```
class Thread1 extends Thread
{
   String x; int t;
   Thread1 (String x, int t)
   {
     this.x = x;
     this.t = t;
    Thread1(String x, int
    public void run()
    {
      try{} {
        while (true)
        {
          System.out.println(x);
          Thread.sleep(t);
        }
       }
```

```
catch (InterruptedException ie)
{
    System.out.println("exit thread");
}
}
}
class Main
{
    psvm
    {
        Thread1 t1 = new Thread1("BMS College",1000);
        Thread1 t2 = new Thread1("CSE", 2000)
        t1.start();
        t2.start();
    }
}
```

Output:
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
.
.
.

28/11/24

```
import java.util.Scanner;
class Thread1  extends Thread
{
/*Thread1(String x)
{
super(x);
System.out.println("First thread "+this);
}*/
public void run()
{
try{
System.out.println("BMS College of Engineering");
Thread.sleep(10000);
}
catch (InterruptedException ie)
{
System.out.println("exit thread");
}
}
}
class Thread2  extends Thread
{
/*Thread2(String y)
{
super(y);
System.out.println("Second thread   "+this);
}*/
public void run()
{
try{
System.out.println("CSE");
```

```
Thread.sleep(2000);
}
catch (InterruptedException ie)
{
System.out.println("exit thread");
}
}
}
class Tdpgm
{
public static void main(String xx[])
{
Thread1 t1=new Thread1(/*"thread 1"*/);
t1.start();
Thread2 t2=new Thread2(/*"thread2"*/);
t2.start();
}
}
```

LABORATORY PROGRAM - 9

9.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.*;
import
java.awt.event.*;

public class DivisionMain1 extends Frame implements ActionListener
{
        TextField
        num1,num2;
        Button dResult;
        Label
        outResult;
        String
        out="";
        double
        resultNum;
        int flag=0;

        public DivisionMain1()
        {
                setLayout(new FlowLayout());

                dResult = new Button("RESULT");
                Label number1 = new Label("Number
                1:",Label.RIGHT); Label number2 = new
                Label("Number 2:",Label.RIGHT);
                num1=new TextField(5);
                num2=new TextField(5);
```

```
        outResult = new Label("Result:",Label.RIGHT);

        add(number1
        );
        add(num1);
        add(number2
        );
        add(num2);
        add(dResult)
        ;
        add(outResul
        t);

        num1.addActionListener(this);
        num2.addActionListener(this);
        dResult.addActionListener(this);
        addWindowListener(new
        WindowAdapter()
        {
                public void windowClosing(WindowEvent we)
                {
                        System.exit(0);
                }
        });
}
public void actionPerformed(ActionEvent ae)
{
        int
        n1,n2;
        try
        {
                if (ae.getSource() == dResult)
                {
                        n1=Integer.parseInt(num1.getText());
                        n2=Integer.parseInt(num2.getText());

                        /*if(n2==0)
                                throw new
                        ArithmeticException();*/ out=n1+"
                        "+n2+" ";
```

```
                    resultNum=n1/n2;
                    out+=String.valueOf(result
                    Num); repaint();

                }
        }
        catch(NumberFormatException e1)
        {
                flag=1;
                out="Number Format Exception!
                "+e1; repaint();
        }
        catch(ArithmeticException e2)
        {
                flag=1;
                out="Divide by 0 Exception!
                "+e2; repaint();
        }

}
public void paint(Graphics g)
{
        if(flag==0)
        g.drawString(out,outResult.getX()+outResult.getWidth(),outR
        esult.getY()+outResult. getHeight()-8);
        else
        g.drawString(out,1
        00,200); flag=0;
}
```

9. 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 & 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.*;
import java.awt.event.*;
public class DivisionMain1 extends Frame implements
                                    ActionListener
{
   TextField
   num1, num2;
   Button dResult;
   Label
   outResult;
   String
   out = "";
   double
   resultNum;
   int flag=0;
   public DivisionMain1()
   {
      setLayout(new FlowLayout());
      dResult = new Button("RESULT");
      label number1= new Label("Number 1:", la
      label number 2 = new Label("Number 2:",
                                    label.RIGHT
      num1 = new TextField(5);
      num2 = new TextField(5);
```

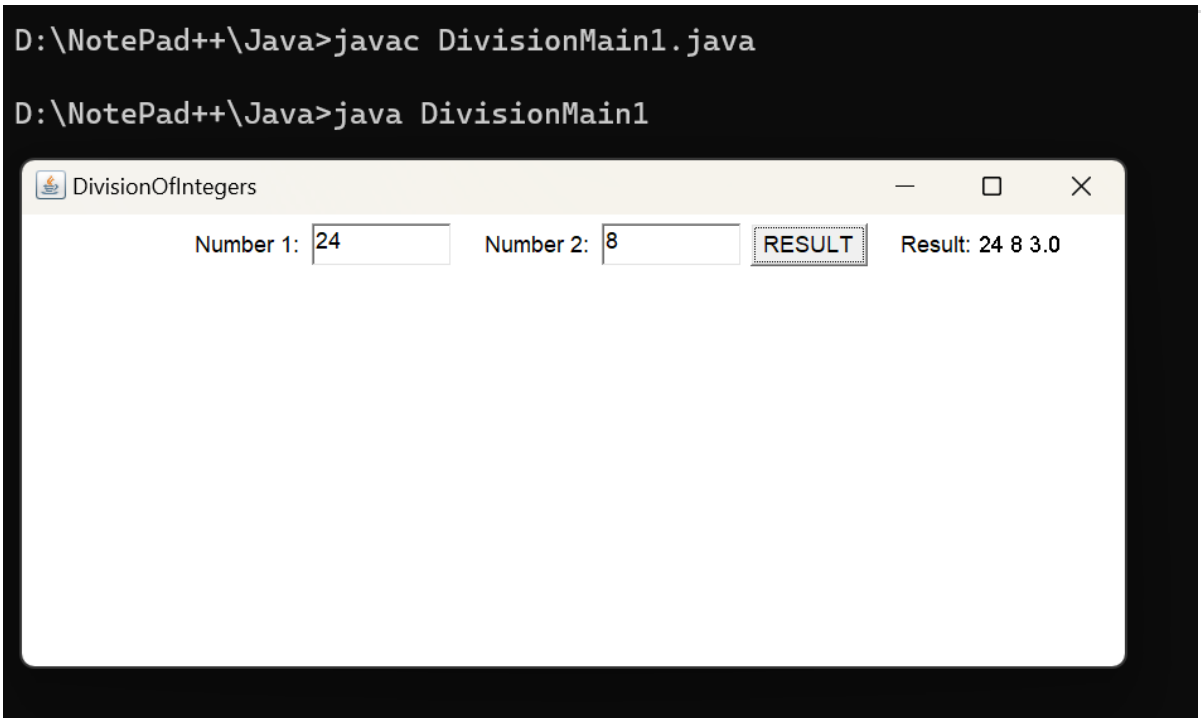
```
outResult = new Label("Result :", Label.RIGHT);
add(number1);
add(num1);
add(number2);
add(dResult);
add(outresult);
num1.addActionListener(this);
num2.addActionListener(this);
dResult.addActionListener(this);
addwindowlistener(new windowAdapter()
{
  public void windowClosing(windowEvent we
  { System.exit(0);
  }
});
}
public void actionPerformed(ActionEvent ae)
{ int n1, n2;
  try
  { if (ae.getSource() == dResult)
    {
      n1 = Integer.parseInt(num1.getText());
      n2 = Integer.parseInt(num2.getText());
    out = n1+" "+n2+" ";
    resultNum = n1/n2;
    out += String.valueOf(resultNum);
    repaint();
  }
}
```

```
catch (NumberFormatException e1)
 {  flag =1;
    out = "Number Format Exception!" + e1;
    Repaint();
 }
}
public void paint (Graphics g)
{
  if (flag==0)
  g.drawString (out, outResult.getX()+ outResult.getW
  outResult.getY()+ OutResult.getHeight()-8);
  else
    g.drawString(out, 100, 200);
    flag=0;
  }
```

Output:

Number 1: 24 Number 2: 8

Result: 24 8 3.0



LAB PROGRAM - 10

10. Demonstrate Interprocess communication and deadlock.

```
class Q {
int n;
boolean valueSet = false;

synchronized int get() {
while(!valueSet)
try {
System.out.println("\nConsumer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
System.out.println("Got: " + n);
valueSet = false;
System.out.println("\nIntimate Producer\n");
notify();
return n;
}

synchronized void put(int n) {
while(valueSet)
try {
System.out.println("\nProducer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n");
notify();
}
}

class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(i<15) {
q.put(i++);
}
}
}

class Consumer implements Runnable {
Q q;
Consumer(Q q) {
this.q = q;
new Thread(this, "Consumer").start();
}
public void run() {
```

```
        int i=0;
while(i<15) {
int r=q.get();
System.out.println("consumed:"+r);
i++;
}
}
}

class PCFixed {
public static void main(String args[]) {
Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop.");
}
}
```

Demonstrate interprocess communication and deadlock.

```
class Q {
int n;
boolean valueSet = false;
synchronized int get(){
while(!valueset)
try{
System.out.println("\n Customer waiting \n");
wait();
}
catch (InterruptedException e) {
System.out.println("InterruptedException caught")
}
System.out.println("Got:"+n);
valueSet = false;
System.out.println("\n Intimate Producer \n");
notify();
return n;
}
synchronized void put(int n){
while (valueset)
try{
System.out.println("\n Producer waiting \n");
wait();
} catch (InterruptedException e){
System.out.println("InterruptedException caught");
}
this.n = n;
Value Set = true;
System.out.println("Put:"+n);
System.out.println("\n Intimate Consumer \n");
notify(); } }
```

```
class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i=0;
while(i<15) {
q.put(i++);
}
}
}
class Consumer implements Runnable{
Q q;
Consumer(Q q){
this.q = q;
new Thread(this, "Consumer").start();
}
public void run(){
  int i=0;
while (i<15) {
int r = q.get();
System.out.println("consumed: "+r);
i++;
} } }
class PCFixed {
public static void main(String args[]){
Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop.");
} }
```

OUTPUT

```
D:\1BM23CS330>java PCFixed
Press Control-C to stop.
Put: 0

Intimate Consumer


Producer waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer


Producer waiting

consumed:0
Got: 1

Intimate Producer

consumed:1
Put: 2

Intimate Consumer


Producer waiting

Got: 2

Intimate Producer

consumed:2
Put: 3

Intimate Consumer


Producer waiting

Got: 3

Intimate Producer

consumed:3
Put: 4

Intimate Consumer


Producer waiting

Got: 4

Intimate Producer

consumed:4
Put: 5

Intimate Consumer
```

```
Producer waiting

Got: 5

Intimate Producer

consumed:5
Put: 6

Intimate Consumer


Producer waiting

Got: 6

Intimate Producer

consumed:6
Put: 7

Intimate Consumer


Producer waiting

Got: 7

Intimate Producer

consumed:7
Put: 8

Intimate Consumer


Producer waiting

Got: 8

Intimate Producer

consumed:8
Put: 9

Intimate Consumer


Producer waiting

Got: 9

Intimate Producer

consumed:9
Put: 10

Intimate Consumer


Producer waiting

Got: 10

Intimate Producer
```

```
consumed:10
Put: 11

Intimate Consumer


Producer waiting

Got: 11

Intimate Producer

consumed:11
Put: 12

Intimate Consumer


Producer waiting

Got: 12

Intimate Producer

consumed:12
Put: 13

Intimate Consumer


Producer waiting

Got: 13

Intimate Producer

consumed:13
Put: 14

Intimate Consumer

Got: 14

Intimate Producer

consumed:14

D:\1BM23CS330>
```

ii. Demonstration of deadlock

```
class A
{
  synchronized void foo(B b)
   { String name = Thread.currentThread().getName();
     System.out.println(name + " entered A.foo");
     try { Thread.sleep(1000); }
     catch(Exception e) { System.out.println("A Interrupted"); }
     System.out.println(name + " trying to call B.last()"); b.last(); }
     synchronized void last() { System.out.println("Inside A.last"); }
}

 class B {
   synchronized void bar(A a) {
   String name = Thread.currentThread().getName();
   System.out.println(name + " entered B.bar");
   try { Thread.sleep(1000); }
catch(Exception e) { System.out.println("B Interrupted"); }
System.out.println(name + " trying to call A.last()"); a.last(); }
synchronized void last() { System.out.println("Inside A.last"); }

}
```

```
class Deadlock implements Runnable
 {
  A a = new A(); B b = new B();
  Deadlock( ) {
    Thread.currentThread().setName("MainThread");
    Thread t = new Thread(this, "RacingThread");
     t.start(); a.foo(b); // get lock on a in this thread.
     System.out.println("Back in main thread");
   }
 public void run() { b.bar(a); // get lock on b in other thread.
  System.out.println("Back in other thread");
  }
public static void main(String args[]) { new Deadlock(); }
}

public static void main(String[] args)
                {
                        DivisionMain1 dm=new DivisionMain1();
                        dm.setSize(new Dimension(800,400));
                        dm.setTitle("DivisionOfIntegers");
                        dm.setVisible(true);
                }
        }
```

demonstration of deadlock

```
class A
{
 synchronized void foo(B b)
  { String name = Thread.currentThread().getName
  System.out.println(name + " entered A.foo");
  try { Thread.sleep(1000); }
  catch (Exception e) { System.out.println("A interrupte
  System.out.println(name + "trying to call B.last()"
   b.last(); }
  synchronized void last()
  {
    System.out.println("Inside A.last"); }
  }
  class B {
  synchronized void bar(A a)
  {
   String name = Thread.currentThread().getName()
   System.out.println(name + "entered B.bar");
   try {
      Thread.sleep(1000); }
  catch(Exception e)
  {
   System.out.println(" B interrupted"); }
  System.out.println(name + "trying to call A.la
   a.last(); }
   synchronized void last() {
       System.out.println("Inside A.last"); }
  }
```

```
class Deadlock implements Runnable
{
    A a = new A();
    B b = new B();
    Deadlock() {
     Thread.currentThread().setName("MainThread");
     Thread t = new Thread(this, "RacingThread");
     t.start(); a.foo(b);
     System.out.println("Back in main thread");
    }
    public void run()
    { b.bar(a);
      System.out.println("Back in other thread");
    }
    public static void main(String args[])
    {
      new Deadlock(); }
    }
    public static void main(String[] args)
     {
        DivisionMain1 dm = new DivisionMain1();
        dm.setSize(new Dimension(800, 400));
       dm.setTitle("DivisionOfIntegers");
       dm.setVisible(true);
       }
     }
```

```
D:\1BM23CS330>javac Deadlock.java

D:\1BM23CS330>java Deadlock
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
MainThread trying to call B.last()
```