VISVESVARAYA TECHNOLOGIVAL UNIVERSITY

"JNANASANGAMA", Belgaum- 590014, Karnataka

LAB REPORT

ON

OBJECT ORIENTED JAVA PROGRAMMING

(23CS3PCOOJ)

Submitted By

DISHA D S

(1BM23CS094)

in partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

BMS COLLEGE OF ENGINEERING

(Autonomous Institute under VTU)

Bengaluru- 560019

Sep- 2024 to 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

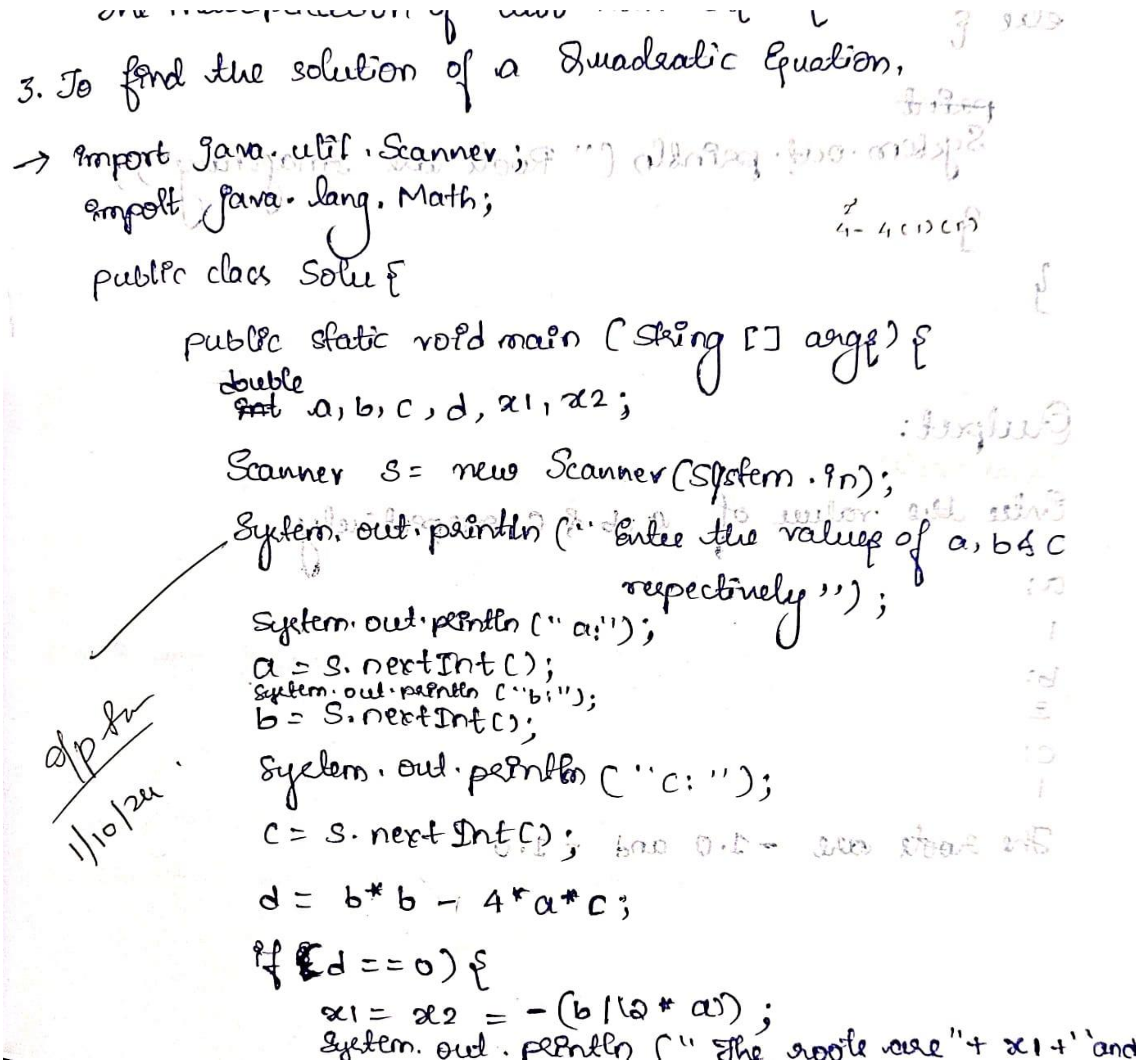
This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **DISHA D S (1BM23CS094),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Lab faculty Incharge Name Assistant Professor Department of CSE, BMSCE	Dr. Kavita Sooda Professor & HOD Department of CSE, BMSCE

Index

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.

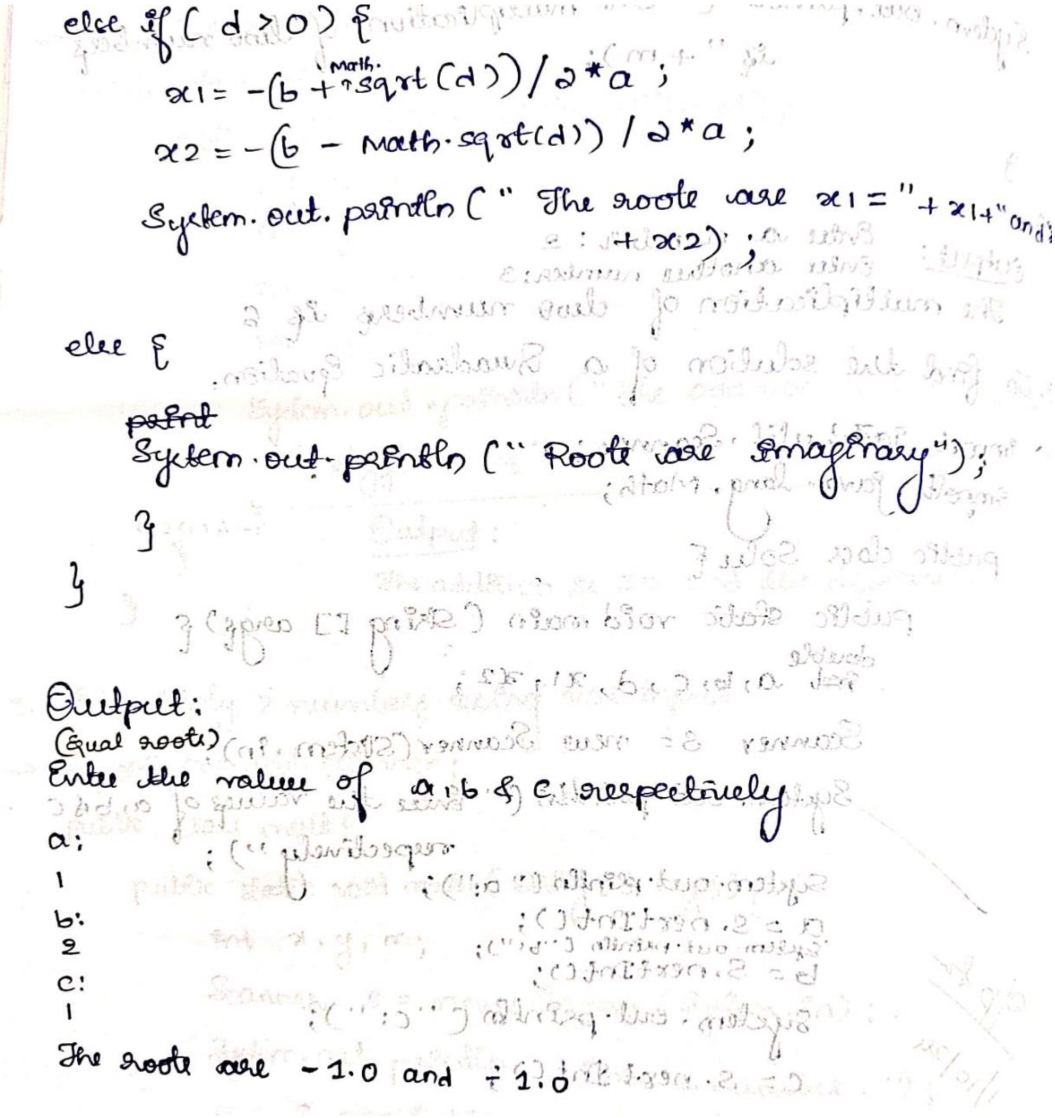


3. To find the solution of a Quadratic Equation,

```
→ import java.util.Scanner;
  import java.lang.Math;
  public class Solu {
        public static void main (String [] args) {
        double
        ~~int~~ a, b, c, d, x1, x2;
        Scanner s= new Scanner(System.in);
        System.out.println (" Enter the values of a, b & c
                                   respectively");
        System.out.println (" a:");
        a = s.nextInt();
        System.out.println ("b:");
        b= S.nextInt();
        System.out.println ("c: ");
        c= s.nextInt();
        d= b* b - 4*a*c;
        if (d ==0) {
            x1 = x2 = -(b/(2* a));
            System.out.println (" The roots are"+ x1+"and
```

4-4(1)(1)

1/10/24



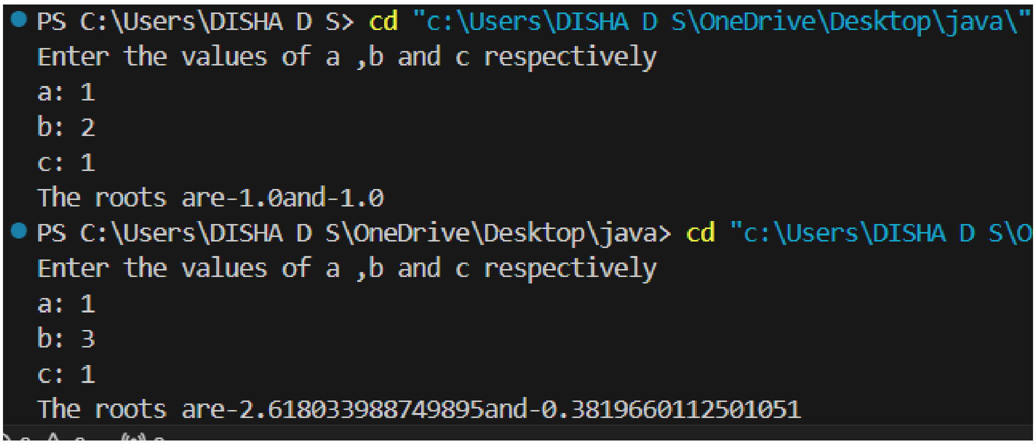
```
else if ( d > 0 ) {
    x1 = -(b + Math.sqrt (d)) / 2*a ;
    x2 = -(b - Math.sqrt(d)) / 2*a ;
    System.out.println (" The roots are x1 = "+ x1+" and"
                                        + x2) ;

else {
    ~~print~~
    System.out.println (" Roots are imaginary");
    }
}
```

Output:
(Equal roots)

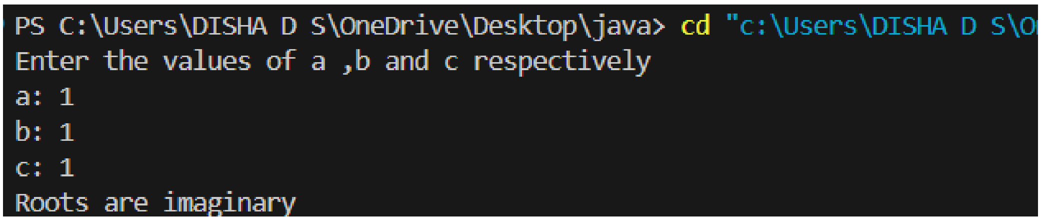
```
Enter the value of a,b & c respectively
a:
1
b:
2
c:
1
The roots are -1.0 and -1.0
```

```
import java.util.Scanner;

public class Solu{
    public static void main(String args[]){
        double a,b,c,d,x1,x2;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the values of a ,b and c respectively ");
        System.out.print("a: ");
        a= s.nextInt();
        System.out.print("b: ");
        b=s.nextInt();
        System.out.print("c: ");
        c= s.nextInt();
        d = b*b-4*a*c;
        if(d==0)
        {
            x1=x2=-(b/(2*a));
            System.out.println("The roots are"+x1+"and"+x2);
        }
        else if(d>0)
        {
            x1=-(b+Math.sqrt(d))/2*a;
            x2=-(b-Math.sqrt(d))/2*a;
            System.out.println("The roots are"+x1+"and"+x2);
        }
        else{
            System.out.println("Roots are imaginary");
        }
    }
}
```

OUTPUT:



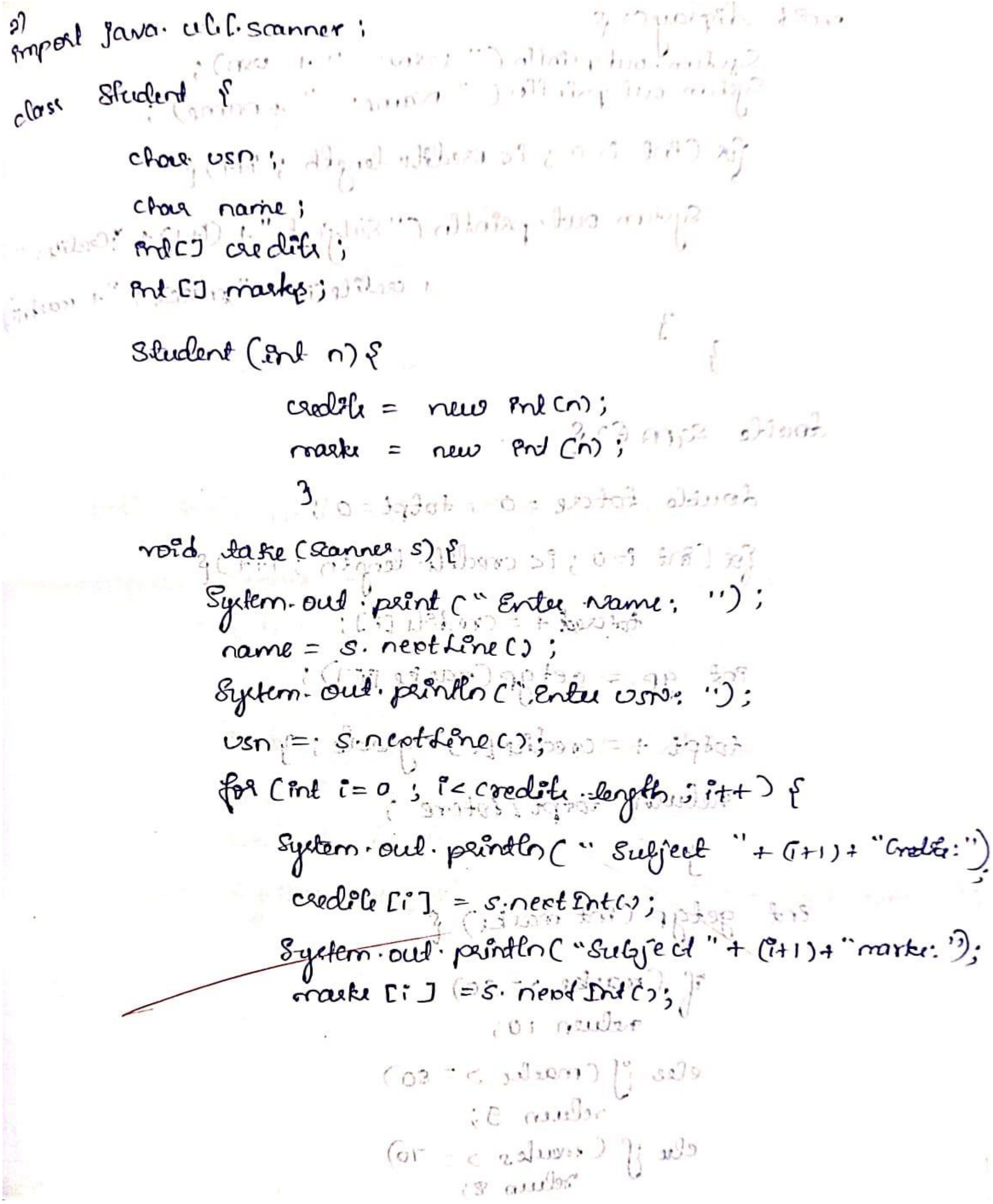
```
PS C:\Users\DISHA D S> cd "c:\Users\DISHA D S\OneDrive\Desktop\java\"
Enter the values of a ,b and c respectively
a: 1
b: 2
c: 1
The roots are-1.0and-1.0
PS C:\Users\DISHA D S\OneDrive\Desktop\java> cd "c:\Users\DISHA D S\O
Enter the values of a ,b and c respectively
a: 1
b: 3
c: 1
The roots are-2.618033988749895and-0.3819660112501051
```



```
PS C:\Users\DISHA D S\OneDrive\Desktop\java> cd "c:\Users\DISHA D S\O
Enter the values of a ,b and c respectively
a: 1
b: 1
c: 1
Roots are imaginary
```

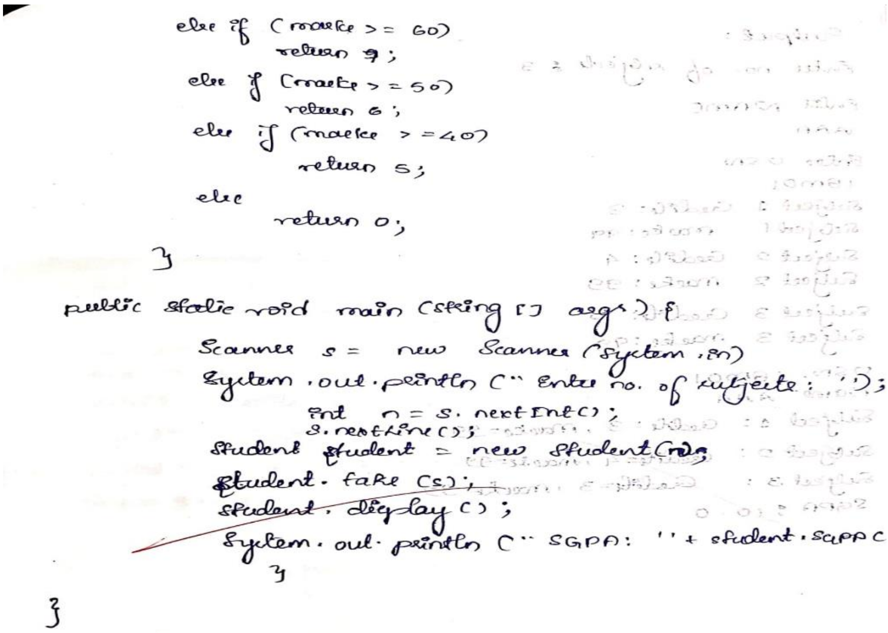
PROGRAM 2

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.

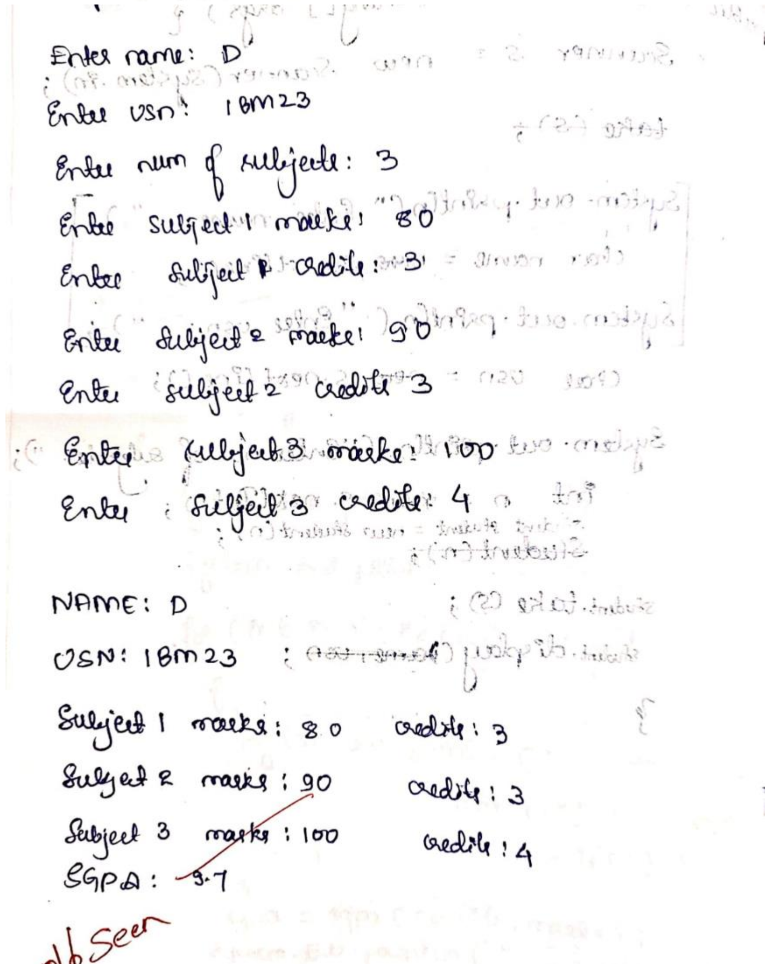


```
2)
import java.util.scanner;

class Student {

        char usn;
        char name;
        int[] credits;
        int [] marks;

        Student (int n) {
                credits = new int(n);
                marks = new int (n);
                }

        void take(Scanner s) {
            System.out.print(" Enter Name: ");
            name = s. nextLine();
            System.out.println(" Enter usn: ");
            usn = s.nextLine();
            for (int i=0 ; i< credits.length ; i++) {
                System.out.println(" Subject "+ (i+1)+ "Credits:")
                credits [i] = s.nextInt();
                System.out.println("Subject "+ (i+1)+"marks: ");
                marks [i] = s. nextInt();
```

```
void display() {
    System.out.println(" USN:  " + usn);
    System.out.println(" Name:  " + name);
    for (int i = 0 ; i< credits.length ; i++) {
        System.out.println(" Subject " + (i+1) + " Credits:
                                + credits[i] + " marks: " + mark
    }
}

double sgpa () {
    double totcre = 0 , totpt = 0 ;
    for (int i=0 ; i< credits.length ; i++) {
        totcred + = credits [i] ;
        int gp = getgp (marks [i]) ;
        totpt + = credits [i] * grade ; }
    return totpt / totcre ;
}

int getgp (int marks) {
    if (marks >= 90)
        return 10;
    else if (marks >= 80)
        return 9;
    else if (marks >= 70)
        return 8
```



```
        else if (marks >= 60)
              return 9;
        else if (marks >= 50)
              return 6;
        else if (marks >= 40)
              return 5;
        else
              return 0;
    }
public static void main (String [] args) {
        Scanner s = new Scanner (System.in)
        System.out.println (" Enter no. of subjects: ");
              int n = s.nextInt();
              s.nextLine();
        Student student = new Student(n);
        student.take (s);
        student.display ();
        System.out.println (" SGPA: " + student.SGPA(
        }
}
```



```
Enter name: D
Enter USN: 1BM23
Enter num of subjects: 3
Enter subject 1 marks: 80
Enter subject 1 credits: 3
Enter subject 2 marks: 90
Enter subject 2 credits 3
Enter subject 3 marks: 100
Enter subject 3 credits 4

NAME: D
USN: 1BM23
Subject 1 marks: 80     credits: 3
Subject 2 marks: 90     credits: 3
Subject 3 marks: 100    credits: 4
SGPA: 9.7
```

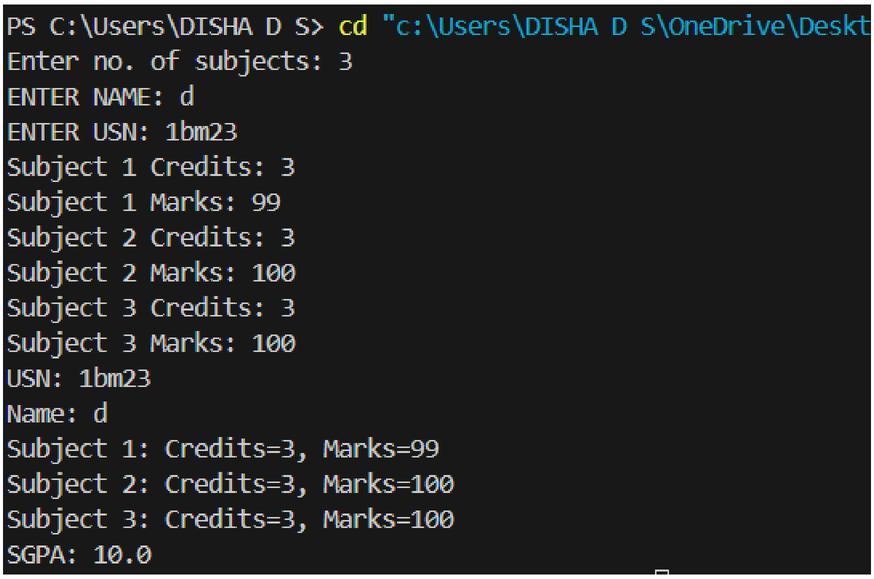
o/p seen

Program:

```
import java.util.Scanner;

class Student {
    String usn;
    String name;
    int[] credits;
    int[] marks;

    Student(int n) {
        credits = new int[n];
        marks = new int[n];
    }



    void take(Scanner s) {
        System.out.println("ENTER NAME: ");
        name = s.nextLine();
        System.out.println();

        System.out.println("ENTER USN: ");
        usn = s.nextLine();

        for (int i = 0; i < credits.length; i++) {
            System.out.print("Subject " + (i + 1) + " Credits: ");
            credits[i] = s.nextInt();
            System.out.print("Subject " + (i + 1) + " Marks: ");
            marks[i] = s.nextInt();

        }
    }



    void display() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);

        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + ": Credits=" +
credits[i] + ", Marks=" + marks[i]);
        }
    }



    double SGPA() {
        double totcre = 0;
        double totpt = 0;
        for (int i = 0; i < credits.length; i++) {
            totcre += credits[i];
            int gp = getgp(marks[i]);
            totpt+= credits[i] * gp;
        }
        return totpt / totcre;
    }


    int getgp(int marks) {
        if (marks >= 90) return 10;
        else if (marks >= 80) return 9;
        else if (marks >= 70) return 8;
        else if (marks >= 60) return 7;
        else if (marks >= 50) return 6;
        else if (marks >= 40) return 5;
        else return 0;
    }

    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter no. of subjects: ");
        int n = s.nextInt();
        s.nextLine();

    Student student = new Student(n);
        student.take(s);
        student.display();
```

```
        System.out.println("SGPA: " + student.SGPA());
    }
}
```

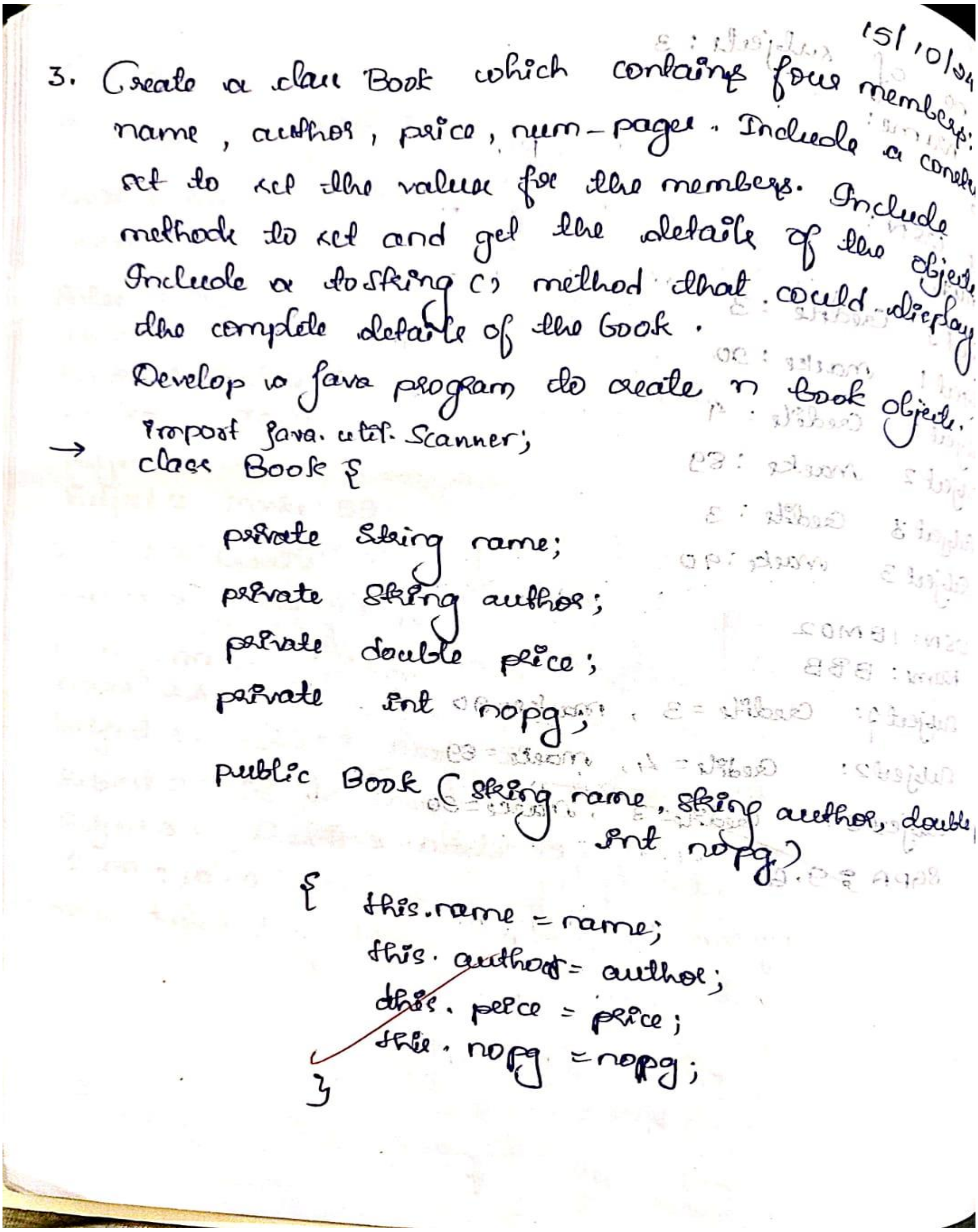
Output:



```
PS C:\Users\DISHA D S> cd "c:\Users\DISHA D S\OneDrive\Deskt
Enter no. of subjects: 3
ENTER NAME: d
ENTER USN: 1bm23
Subject 1 Credits: 3
Subject 1 Marks: 99
Subject 2 Credits: 3
Subject 2 Marks: 100
Subject 3 Credits: 3
Subject 3 Marks: 100
USN: 1bm23
Name: d
Subject 1: Credits=3, Marks=99
Subject 2: Credits=3, Marks=100
Subject 3: Credits=3, Marks=100
SGPA: 10.0
```

Program 3

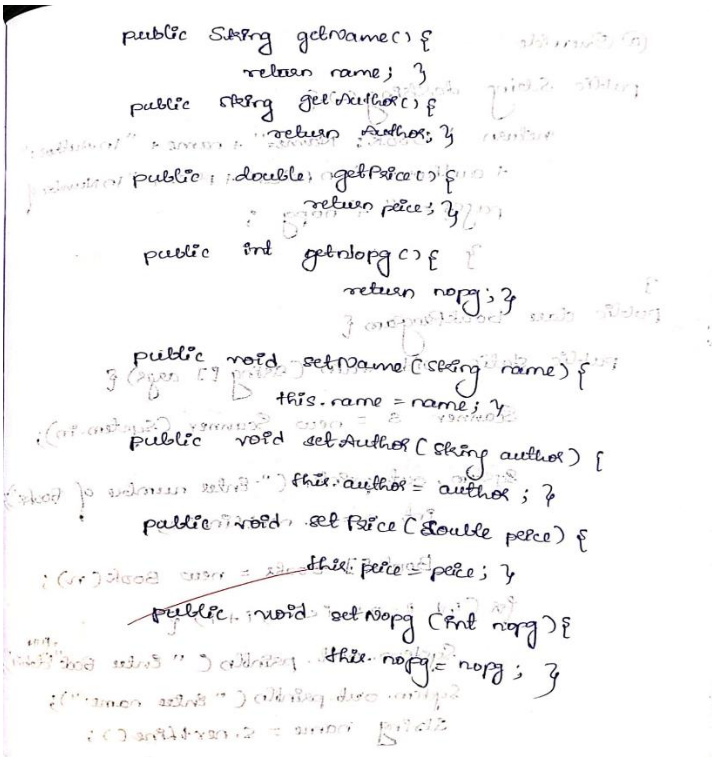
Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program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



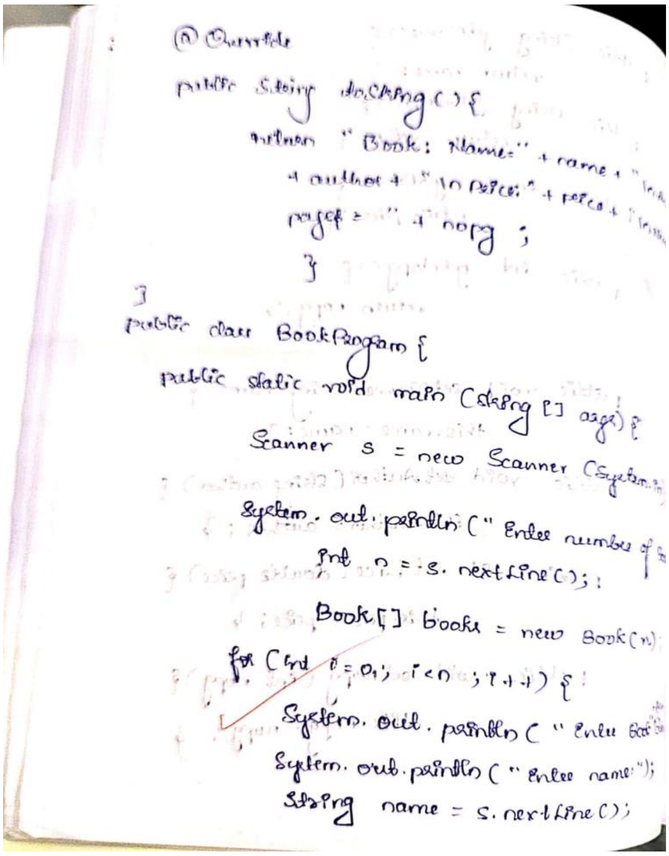
15/10/24

3. Create a class Book which contains four members: name , author , price , num-pages . Include a const set 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.

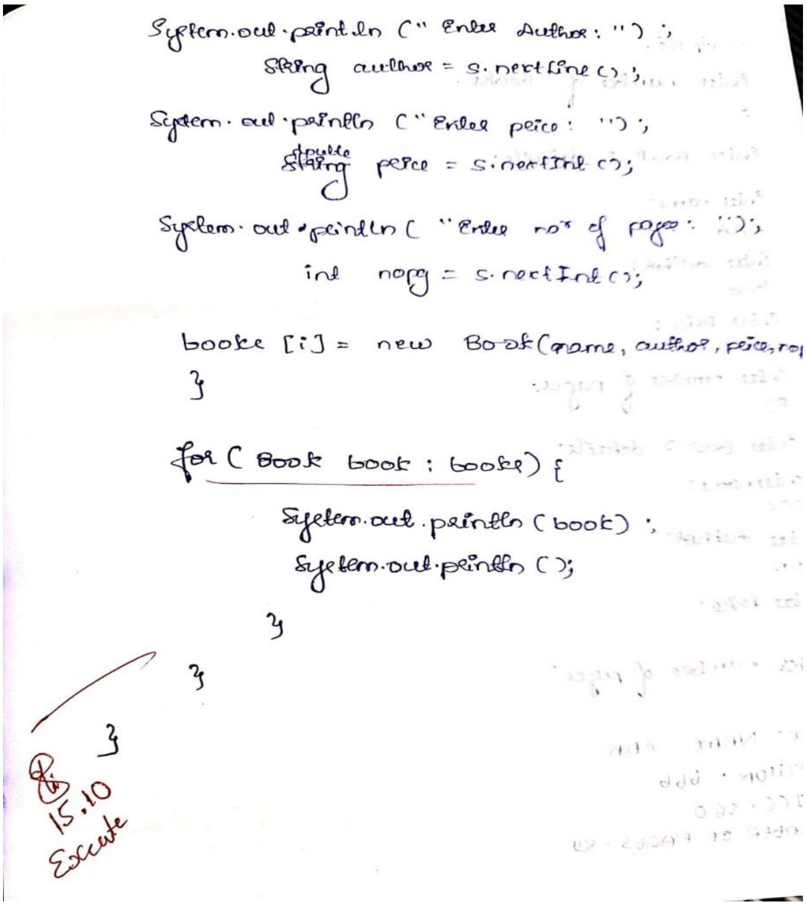
→
```
import java.util.Scanner;
class Book {

    private String name;
    private String author;
    private double price;
    private int nopg;

    public Book (String name, String author, double
                 int nopg)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.nopg = nopg;
    }
```



```
public String getName() {
                return name; }
public String getAuthor() {
                return Author; }
public double getPrice() {
                return peice; }
public int getNopg() {
                return nopg; }

public void setName(String name) {
                this.name = name; }
public void setAuthor(String author) {
                this.author = author; }
public void setPrice(double peice) {
                this.peice = peice; }
public void setNopg(int nopg) {
                this.nopg = nopg; }
```

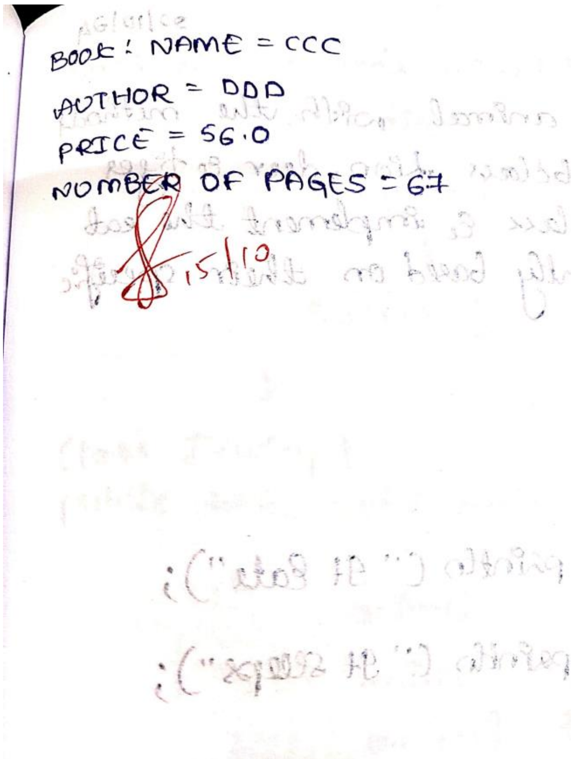


```
@Override
public String toString() {
        return "Book: Name:" + name + "
        + author + "\n Price:" + peice + "
        pages =" + nopg ;
        }
}
public class BookProgram {
  public static void main (String [] args) {
        Scanner s = new Scanner (System.in
        System.out.println(" Enter number of
              int n = s.nextLine();
              Book[] books = new Book(n);
    for (int i = 0; i<n ; i++) {
        System.out.println( " Enter Book
        System.out.println( " Enter name:");
        String name = s.nextLine();
```



```
System.out.println (" Enter Author: ") ;
        String author = s.nextLine ();;
System.out.println (" Enter price: ") ;
        double
        ~~String~~ price = s.nextInt ();
System.out.println ( "Enter no's of pages: ");
        int nopg = s.nextInt ();

  books [i] = new Book(name, author, price, nop
  }

  for ( Book book : books) {
        System.out.println (book) ;
        System.out.println ();

    }

  }

}
```

15.10
Execute



BOOK: NAME = CCC
AUTHOR = DDD
PRICE = 56.0
NUMBER OF PAGES = 67

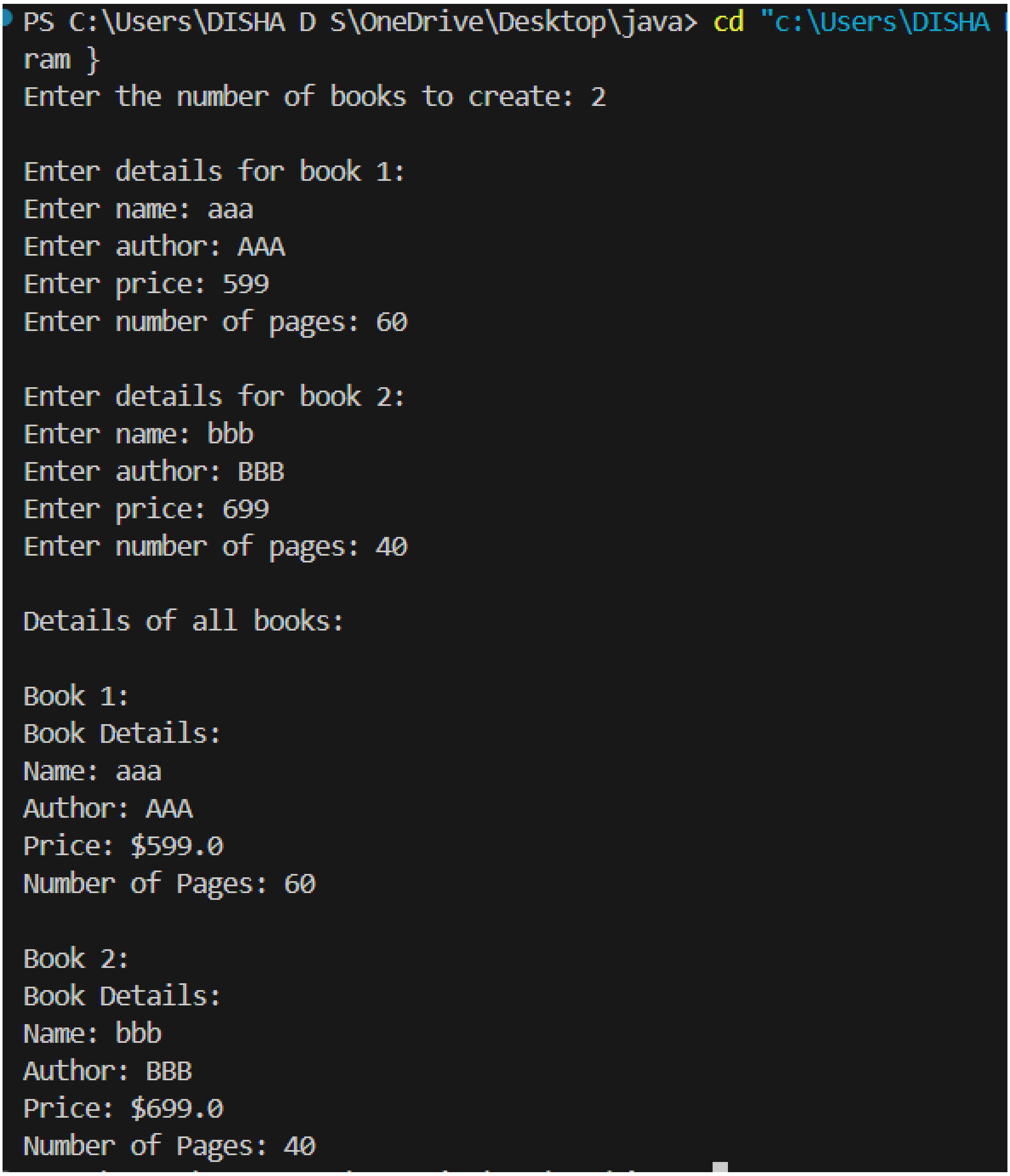
15/10

Program:

```
import java.util.Scanner;



class Book{
   private String name;
   private String author;
   private double price;
   private int numPages;



   public Book(String name, String author, double price, int
numPages) {
      this.name = name;
      this.author = author;
      this.price = price;
      this.numPages = numPages;
   }



   public void setName(String name) {
      this.name = name;
   }

   public void setAuthor(String author) {
    this.author = author;
}

public void setPrice(double price) {
    this.price = price;
}

public void setNumPages(int numPages) {
    this.numPages = numPages;
}



public String getName() {
    return name;
}

public String getAuthor() {
    return author;
}

public double getPrice() {
    return price;
}

public int getNumPages() {
        return numPages;
    }


    @Override
    public String toString() {
        return "Book Details:\n" +
                "Name: " + name + '\n' +
                "Author: " + author + '\n' +
                "Price: $" + price + '\n' +
                "Number of Pages: " + numPages;
    }
}

public class BookProgram{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of books to create: ");
        int n = scanner.nextInt();
        scanner.nextLine();

        Book[] books = new Book[n];
```

```
for (int i = 0; i < n; i++) {
    System.out.println("\nEnter details for book " + (i + 1) + ":");
    System.out.print("Enter name: ");
    String name = scanner.nextLine();

    System.out.print("Enter author: ");
    String author = scanner.nextLine();

    System.out.print("Enter price: ");
    double price = scanner.nextDouble();

    System.out.print("Enter number of pages: ");
    int numPages = scanner.nextInt();
    scanner.nextLine();


    books[i] = new Book(name, author, price, numPages);
}


System.out.println("\nDetails of all books:");
for (int i = 0; i < n; i++) {
    System.out.println("\nBook " + (i + 1) + ":");
    System.out.println(books[i].toString());
}
```

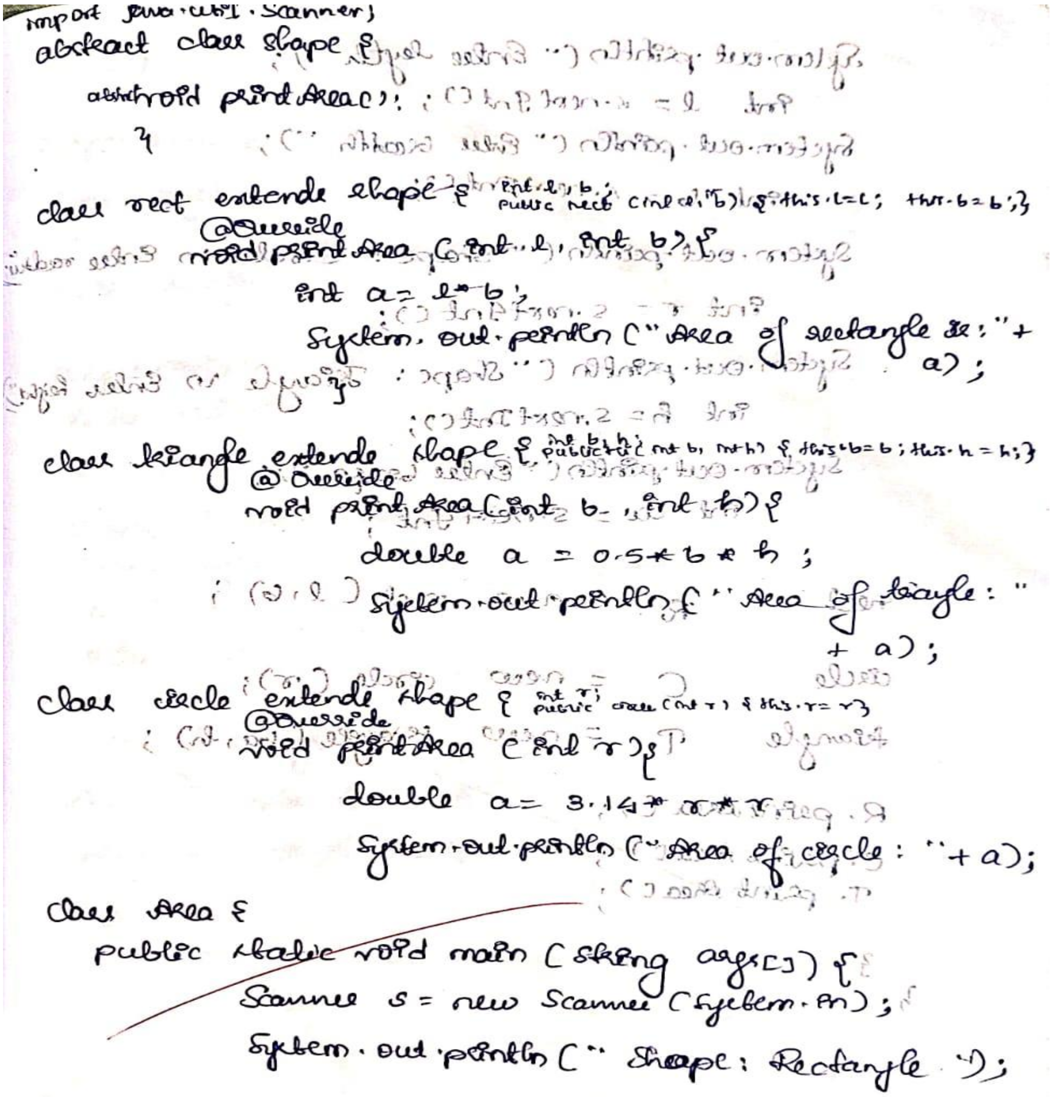
```
        scanner.close();
    }
}
```



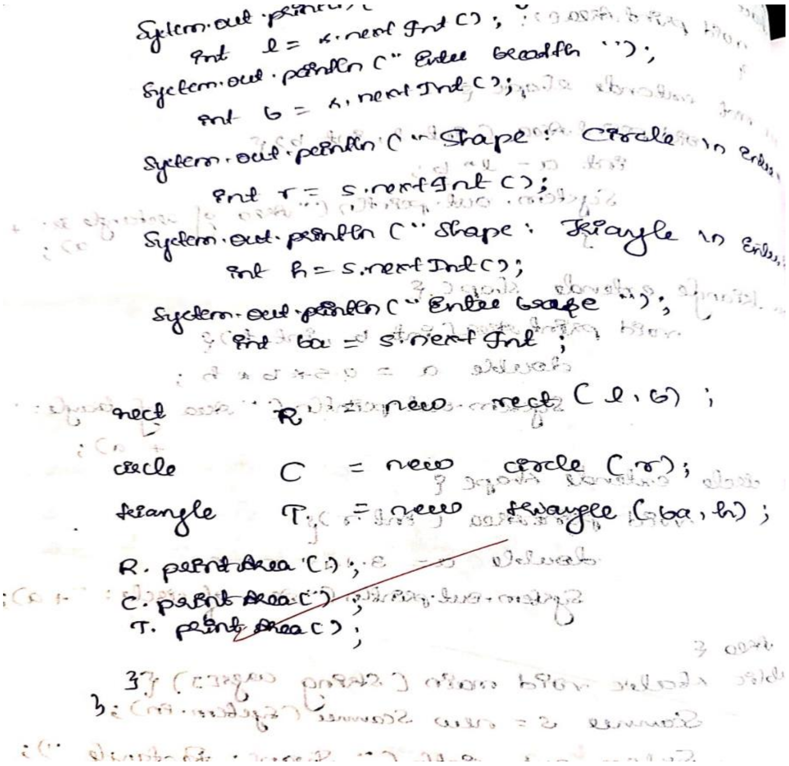
```
PS C:\Users\DISHA D S\OneDrive\Desktop\java> cd "c:\Users\DISHA
ram }
Enter the number of books to create: 2

Enter details for book 1:
Enter name: aaa
Enter author: AAA
Enter price: 599
Enter number of pages: 60

Enter details for book 2:
Enter name: bbb
Enter author: BBB
Enter price: 699
Enter number of pages: 40

Details of all books:

Book 1:
Book Details:
Name: aaa
Author: AAA
Price: $599.0
Number of Pages: 60

Book 2:
Book Details:
Name: bbb
Author: BBB
Price: $699.0
Number of Pages: 40
```

Program 4

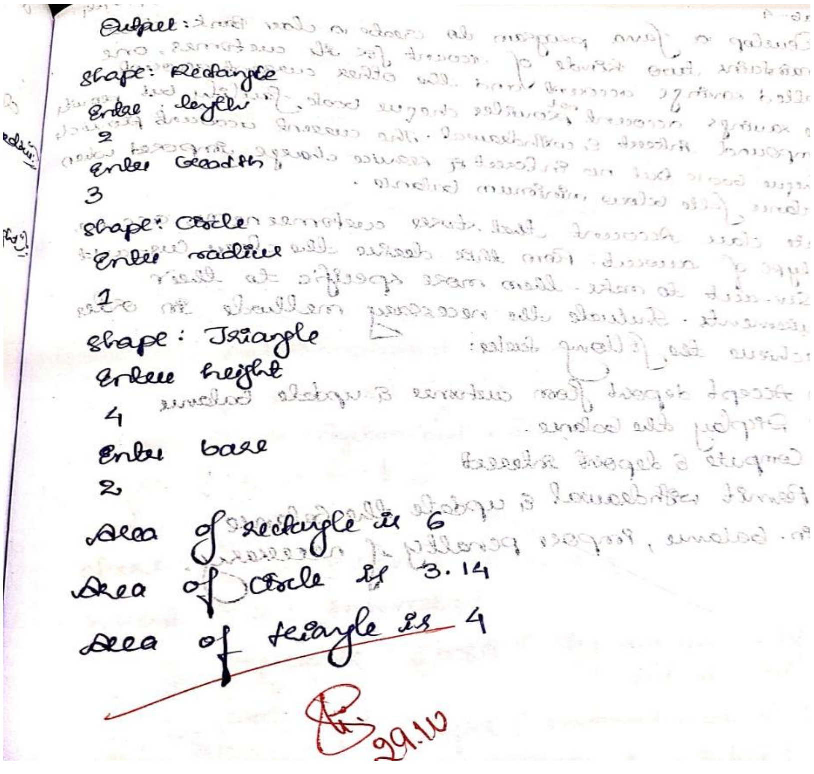
Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea(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



```
import java.util.Scanner;
abstract class shape {
    abstract void printArea();
    }

class rect extends shape { int l, b; public rect(int l, int b){ this.l=l; this.b=b;}
       @Override
       void printArea (int l, int b) {
              int a= l*b;
              System.out.println ("Area of rectangle is: "+
                                                        a);

class triangle extends shape { int b, h; public triangle(int b, int h) { this.b=b; this.h=h;}
       @Override
       void printArea(int b , int h) {
              double a = 0.5*b*h ;
              System.out.println(" Area of triangle: "
                                                        + a);

class circle extends shape { int r; public circle(int r) { this.r=r}
       @Override
       void printArea (int r) {
              double a= 3.14*r*r;
              System.out.println ("Area of circle : "+ a);

class Area {
   public static void main (String args[]) {
       Scanner s = new Scanner (System.in);
       System.out.println (" Shape: Rectangle ");
```



```
System.out.println(
    int l = s.nextInt();
System.out.println("Enter breadth ");
    int b = s.nextInt();
System.out.println("\nShape: Circle\n Enter
    int r = s.nextInt();
System.out.println("Shape: Triangle \n Enter
    int h = s.nextInt();
System.out.println("Enter base ");
    int ba = s.nextInt;

rect        R = new rect (l, b);
circle      C = new circle (r);
triangle    T = new triangle (ba, h);

R.printArea();
C.printArea();
T.printArea();

}}
}
```



```
Output:

Shape: Rectangle
Enter length
2
Enter breadth
3
Shape: Circle
Enter radius
1
Shape: Triangle
Enter height
4
Enter base
2
Area of rectangle is 6
Area of circle is 3.14
Area of triangle is 4
```

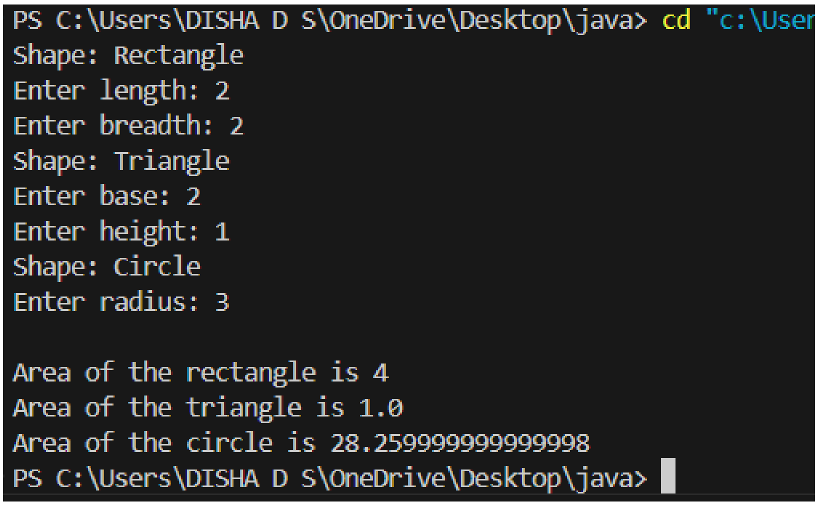
29.10

```
import java.util.Scanner;

abstract class Shape {
   abstract void printArea();
}

class Rect extends Shape {
   private int length;
   private int breadth;

   public Rect(int length, int breadth) {
      this.length = length;
      this.breadth = breadth;
   }

   @Override
   void printArea() {
      int area = length * breadth;
      System.out.println("Area of the rectangle is " + area);
   }
}

class Tri extends Shape {
   private int base;
   private int height;
```

```
    public Tri(int base, int height) {
        this.base = base;
        this.height = height;
    }

    @Override
    void printArea() {
        double area = 0.5 * base * height;
        System.out.println("Area of the triangle is " + area);
    }
}

class Circle extends Shape {
    private int radius;

    public Circle(int radius) {
        this.radius = radius;
    }

    @Override
    void printArea() {
        double area = 3.14 * radius * radius;
        System.out.println("Area of the circle is " + area);
    }
```

```
}

public class Area {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Shape: Rectangle\nEnter length: ");
        int length = scanner.nextInt();
        System.out.print("Enter breadth: ");
        int breadth = scanner.nextInt();

        System.out.print("Shape: Triangle\nEnter base: ");
        int base = scanner.nextInt();
        System.out.print("Enter height: ");
        int height = scanner.nextInt();

        System.out.print("Shape: Circle\nEnter radius: ");
        int radius = scanner.nextInt();

        Rect rectangle = new Rect(length, breadth);
        Tri triangle = new Tri(base, height);
        Circle circle = new Circle(radius);

        System.out.println();
        rectangle.printArea();
        triangle.printArea();
        circle.printArea();


        scanner.close();
    }
}
```

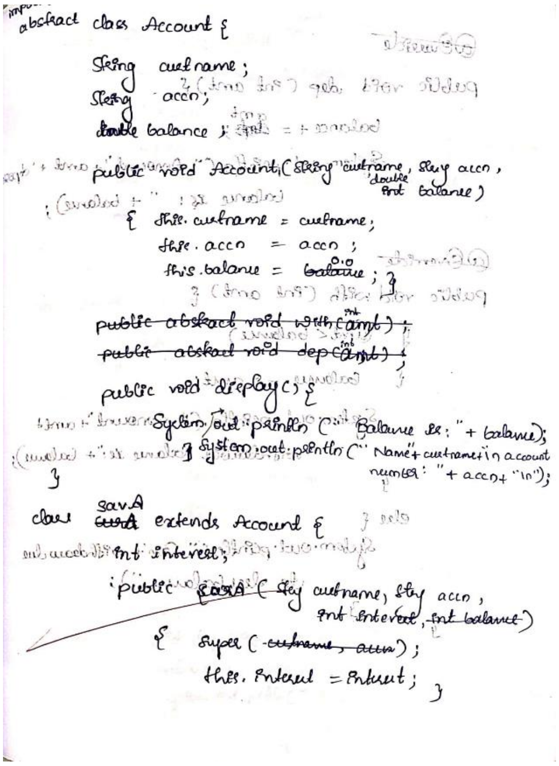
OUTPUT:



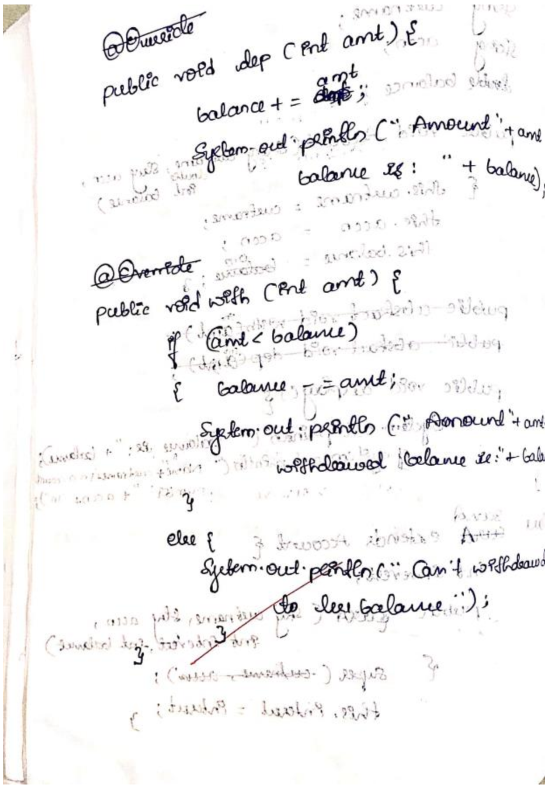
```
PS C:\Users\DISHA D S\OneDrive\Desktop\java> cd "c:\User
Shape: Rectangle
Enter length: 2
Enter breadth: 2
Shape: Triangle
Enter base: 2
Enter height: 1
Shape: Circle
Enter radius: 3

Area of the rectangle is 4
Area of the triangle is 1.0
Area of the circle is 28.259999999999998
PS C:\Users\DISHA D S\OneDrive\Desktop\java>
```

PROGRAM 5

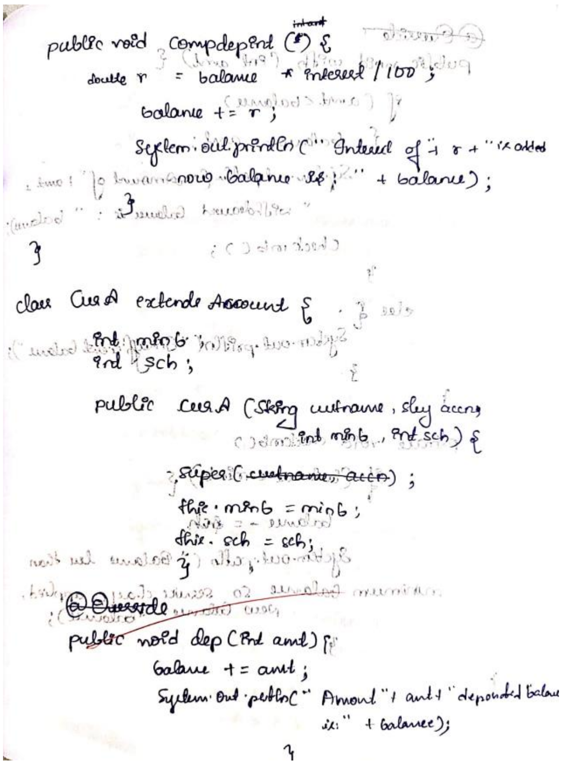
Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book 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.



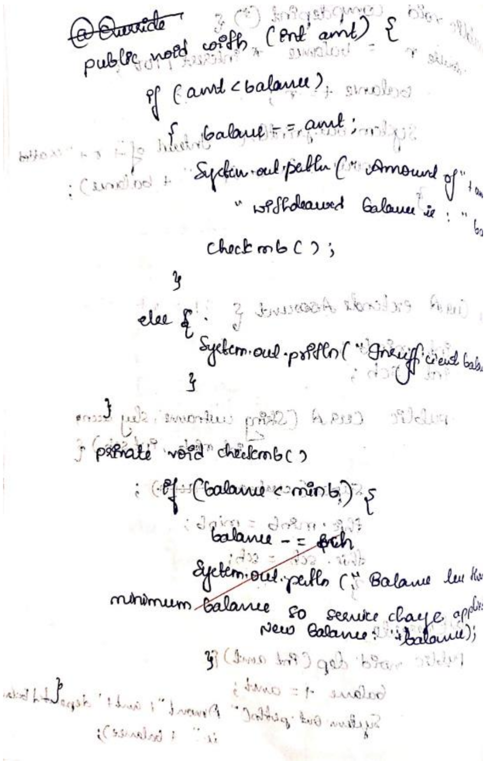
```
impo
abstract class Account {
        String custname;
        String accn;
        double balance ;

    public void Account(String custname, String accn,
                                      double
                                      int balance )
          { this.custname = custname;
            this.accn   = accn ;
                                0.0
            this.balance = balance ; }

        ~~public abstract void with(int amt) ;~~
        ~~public abstract void dep(int amt) ;~~

        public void display() {
          System.out.println(" Balance is: "+ balance);
        } System.out.println(" Name"+ custname+"in account
                                           number: "+ accn+ "\n");
   }
          savA
class ~~curA~~ extends Account {
       int interest;
        public ~~savA~~ ( String custname, String accn,
                             int interest, ~~int balance~~)
           {  super( ~~custname, accn~~);
              this. interest = interest;    }
```



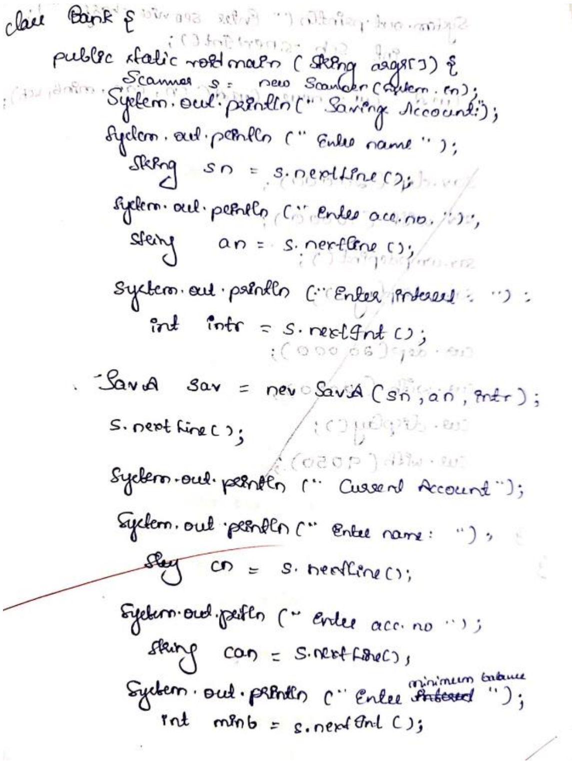
```
~~@Override~~
public void dep (int amt) {
              balance += amt;
          System.out.println(" Amount "+ amt
                         balance is:  " + balance);

@Override
public void with (int amt) {
        if (amt < balance)
        {   balance -= amt;
            System.out.println(" Amount "+ amt
                          withdrawal balance is:"+ bala
        }
        else {
          System.out.println(" Can't withdraw
                        to less balance ");
        }
     }
```



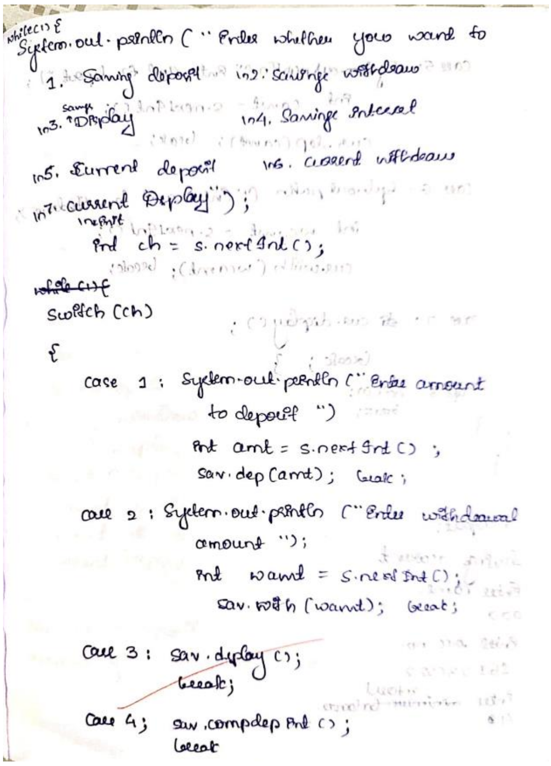
```
public void compdepint (~~i~~) {
              intrest
    double r = balance * interest / 100;
        balance += r;
        System.out.println(" Interest of "+ r + " is added
        now Balance is : " + balance);
    }
}

class CurA extends Account {
    int minb ;
    int sch ;

    public CurA (String custname, String accno
                  int minb , int sch) {

        super(~~custname, accn~~) ;
        this.minb = minb ;
        this.sch = sch;
    }

~~@Override~~
public void dep (int amt) {
        balance += amt ;
        System.out.println(" Amount "+ amt+ " deposited balance
                            is: " + balance);
    }
```



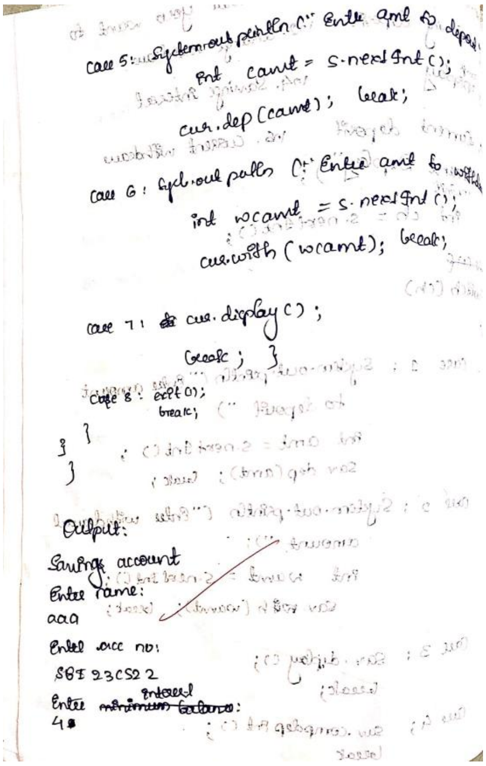
```
~~@Override~~
public void with (int amt) {
    if (amt < balance)
    {   balance -= amt ;
        System.out.println (" Amount of " + am
          " withdrawed Balance is : " ba

        checkmb ( ) ;
    }
    else {
        System.out.println(" Insufficient bal
    }
}
private void checkmb( )
    if (balance < minb) {
        balance -= sch
        System.out.println (" Balance less tha
minimum balance so service charge appli
        New Balance : " +balance);
    }
}
```



```
class Bank {
    public static void main ( String args[]) {
        Scanner s = new Scanner (System.in);
        System.out.println(" Savings Account");
        System.out.println (" Enter name ");
          String   sn  =  s.nextLine ();
        System.out.println (" Enter acc.no. ");
          String    an  =  s.nextLine ();

        System.out.println (" Enter interest : ") ;
            int   intr  =  s.nextInt ();

        SavA  sav  =  new SavA (sn, an, intr);
        s.nextLine ();

        System.out.println (" Current Account ");
        System.out.println (" Enter name: ") ;
          String   cn  =  s.nextLine ();

        System.out.println (" Enter acc.no ");
          String  can  =  s.nextLine(),
                                   minimum balance
        System.out.println (" Enter ~~interest~~ ");
          int   minb  =  s.nextInt ();
```



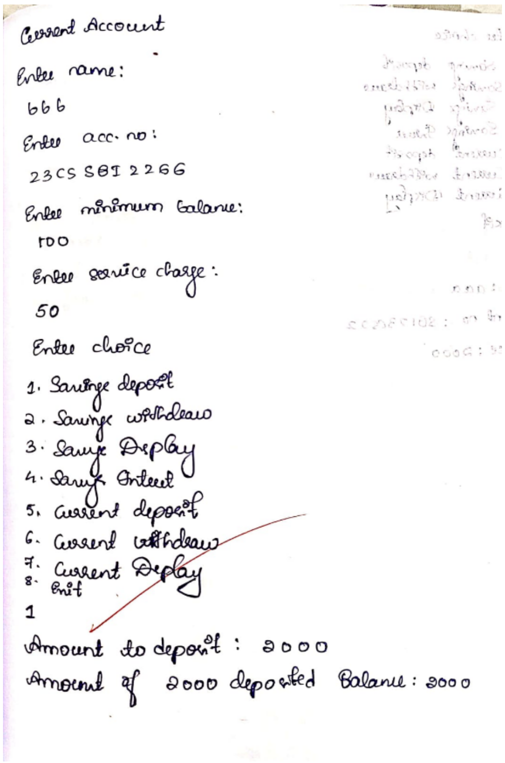
```
while(1) {
System.out.println ( " Enter whether you want to
  1. Saving deposit    \n2. Savings withdraw
       Savings
\n3. Display           \n4. Savings Interest
\n5. Current deposit   \n6. current withdraw
\n7. current Display ") ;
        \nexit
     int  ch =  s.nextInt ();
~~while (1) {~~
 switch (ch)
 {
    case  1 :  System.out.println (" Enter amount
                    to deposit  ")
                int  amt = s.nextInt ()  ;
                sav.dep (amt);  break ;
    case  2 :  System.out.println (" Enter withdrawal
                amount  ");
                int   wamt  =  s.nextInt ();
                  sav.with (wamt);  break;

    case 3 :   sav.display ();
               break;

    case 4;    sav.compdepint () ;
               break
```



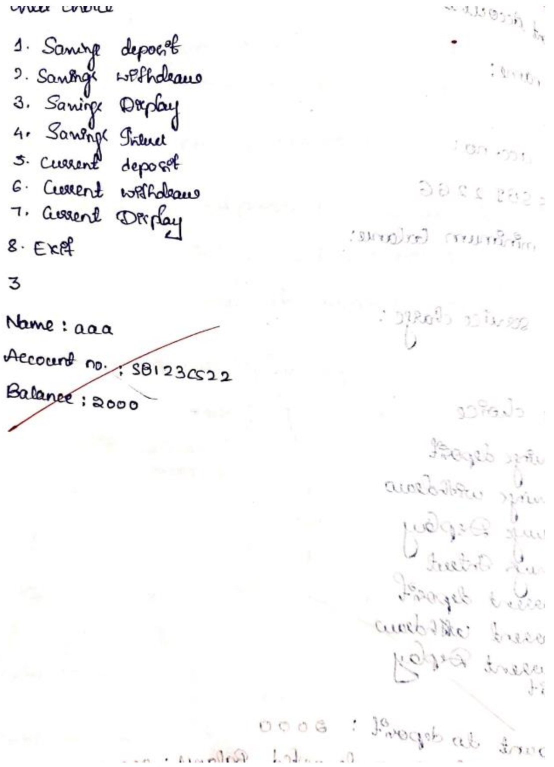
```
case 5: System.out.println(" Enter amt to deposit
        int camt = s.nextInt();
        cur.dep(camt); break;
case 6: syst.out.println (" Enter amt to withdraw
        int wcamt = s.nextInt();
        cur.with (wcamt); break;

case 7: cur.display();
        break; }
case 8: exit(0);
        break;
} }
}
```

Output:

```
Savings account
Enter name:
aaa
Enter acc no:
SBI 23CS22
Enter ~~minimum balance~~ Interest:
4
```



```
Current Account
Enter name:
666
Enter acc. no:
23CS SBI 2266
Enter minimum balance:
100
Enter service charge:
50
Enter choice
1. Savings deposit
2. Savings withdraw
3. Savings Display
4. Savings Interest
5. Current deposit
6. Current withdraw
7. Current Display
8. Exit
1
Amount to deposit : 2000
Amount of 2000 deposited Balance : 2000
```

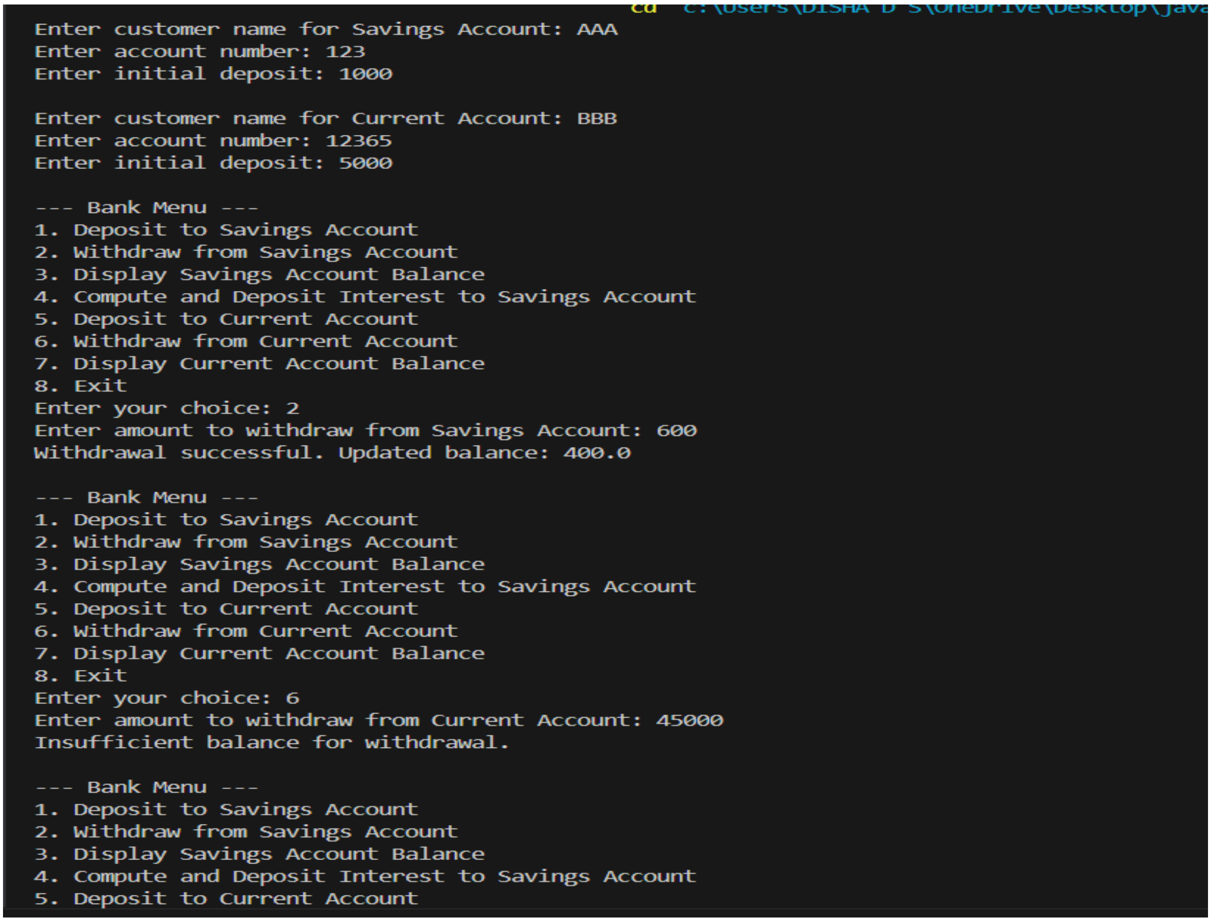


```
Enter choice
1. Savings deposit
2. Savings withdraw
3. Savings Display
4. Savings Interest
5. Current deposit
6. Current withdraw
7. Current Display
8. Exit
3
Name : aaa
Account no. : SBI23CS22
Balance : 2000
```

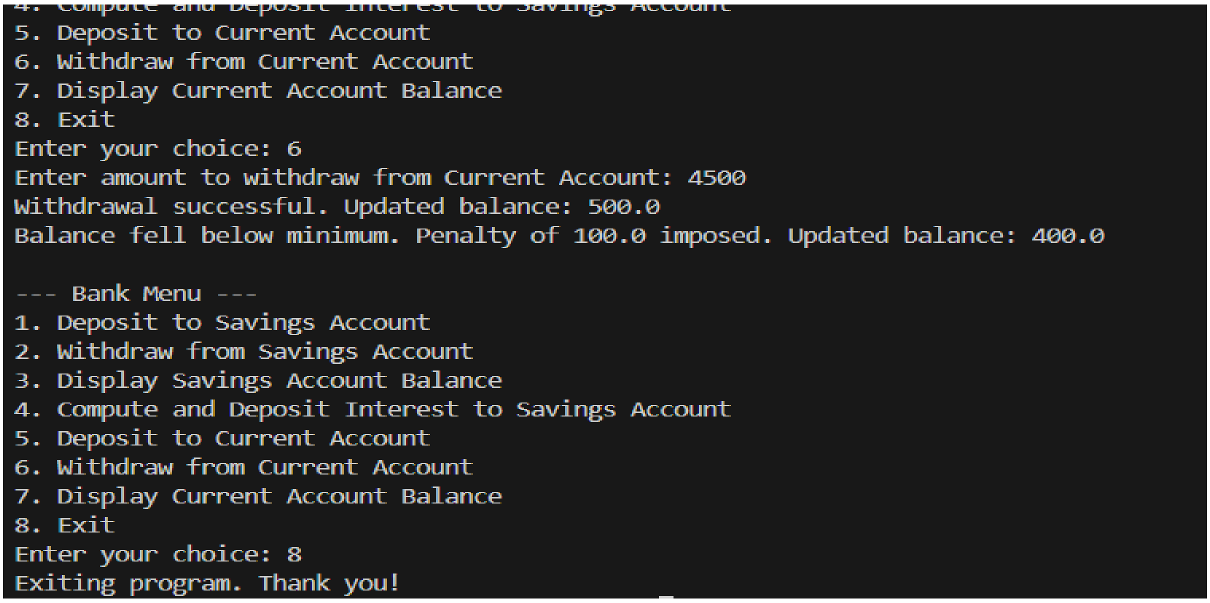
```
import java.util.Scanner;


class Account {
    String customerName;
    String accountNumber;
    String accountType;
    double balance;


    public Account(String customerName, String accountNumber,
String accountType, double initialDeposit) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = initialDeposit;
    }


    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful. Updated balance: " +
balance);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }
    public void displayBalance() {
        System.out.println("Balance: " + balance);
    }
}

class SavAcct extends Account {
    private static final double INTEREST_RATE = 0.04;

    public SavAcct(String customerName, String accountNumber,
double initialDeposit) {
        super(customerName, accountNumber, "Savings",
initialDeposit);
    }

    public void computeAndDepositInterest() {
        double interest = balance * INTEREST_RATE;
        balance += interest;
        System.out.println("Interest of " + interest + " deposited.
Updated balance: " + balance);
    }

    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance for withdrawal.");
        } else {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: 
" + balance);
        }
    }
}

class CurAcct extends Account {
    private static final double MINIMUM_BALANCE = 1000.0;
    private static final double PENALTY = 100.0;

    public CurAcct(String customerName, String accountNumber, 
double initialDeposit) {
        super(customerName, accountNumber, "Current", 
initialDeposit);
    }

    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance for withdrawal.");
        } else {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: 
" + balance);
            if (balance < MINIMUM_BALANCE) {
                imposePenalty();
            }
        }
    }

    private void imposePenalty() {
        balance -= PENALTY;
        System.out.println("Balance fell below minimum. Penalty of " + PENALTY + " imposed. Updated balance: " + balance);
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);



        System.out.print("Enter customer name for Savings Account: ");
        String savName = scanner.nextLine();
        System.out.print("Enter account number: ");
        String savAccNum = scanner.nextLine();
        System.out.print("Enter initial deposit: ");
        double savDeposit = scanner.nextDouble();
        SavAcct savings = new SavAcct(savName, savAccNum, savDeposit);
        scanner.nextLine();
        System.out.print("\nEnter customer name for Current Account:
");
        String curName = scanner.nextLine();
        System.out.print("Enter account number: ");
        String curAccNum = scanner.nextLine();
        System.out.print("Enter initial deposit: ");
        double curDeposit = scanner.nextDouble();
        CurAcct current = new CurAcct(curName, curAccNum,
curDeposit);

        while (true) {
            System.out.println("\n--- Bank Menu ---");
            System.out.println("1. Deposit to Savings Account");
            System.out.println("2. Withdraw from Savings Account");
            System.out.println("3. Display Savings Account Balance");
            System.out.println("4. Compute and Deposit Interest to
Savings Account");
            System.out.println("5. Deposit to Current Account");
            System.out.println("6. Withdraw from Current Account");
            System.out.println("7. Display Current Account Balance");
            System.out.println("8. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
```

```
      switch (choice) {
        case 1:
          System.out.print("Enter amount to deposit to Savings
Account: ");
          double savDepositAmt = scanner.nextDouble();
          savings.deposit(savDepositAmt);
          break;
        case 2:
          System.out.print("Enter amount to withdraw from
Savings Account: ");
          double savWithdrawAmt = scanner.nextDouble();
          savings.withdraw(savWithdrawAmt);
          break;
        case 3:
          System.out.println("Savings Account Balance: ");
          savings.displayBalance();
          break;
        case 4:
          savings.computeAndDepositInterest();
          break;
        case 5:
          System.out.print("Enter amount to deposit to Current
Account: ");
          double curDepositAmt = scanner.nextDouble();
          current.deposit(curDepositAmt);
```

```
                    break;
                case 6:
                    System.out.print("Enter amount to withdraw from
Current Account: ");
                    double curWithdrawAmt = scanner.nextDouble();
                    current.withdraw(curWithdrawAmt);
                    break;
                case 7:
                    System.out.println("Current Account Balance: ");
                    current.displayBalance();
                    break;
                case 8:
                    System.out.println("Exiting program. Thank you!");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }
    }
}
```



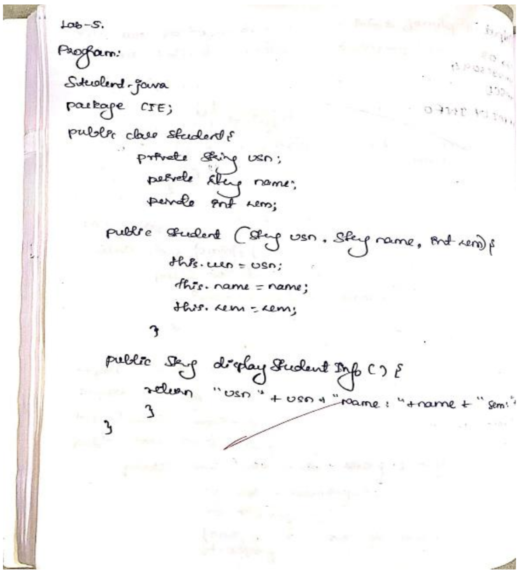
```
                                                    cd  c:\Users\DISHA D S\OneDrive\Desktop\java
  Enter customer name for Savings Account: AAA
  Enter account number: 123
  Enter initial deposit: 1000

  Enter customer name for Current Account: BBB
  Enter account number: 12365
  Enter initial deposit: 5000

  --- Bank Menu ---
  1. Deposit to Savings Account
  2. Withdraw from Savings Account
  3. Display Savings Account Balance
  4. Compute and Deposit Interest to Savings Account
  5. Deposit to Current Account
  6. Withdraw from Current Account
  7. Display Current Account Balance
  8. Exit
  Enter your choice: 2
  Enter amount to withdraw from Savings Account: 600
  Withdrawal successful. Updated balance: 400.0

  --- Bank Menu ---
  1. Deposit to Savings Account
  2. Withdraw from Savings Account
  3. Display Savings Account Balance
  4. Compute and Deposit Interest to Savings Account
  5. Deposit to Current Account
  6. Withdraw from Current Account
  7. Display Current Account Balance
  8. Exit
  Enter your choice: 6
  Enter amount to withdraw from Current Account: 45000
  Insufficient balance for withdrawal.

  --- Bank Menu ---
  1. Deposit to Savings Account
  2. Withdraw from Savings Account
  3. Display Savings Account Balance
  4. Compute and Deposit Interest to Savings Account
  5. Deposit to Current Account
```



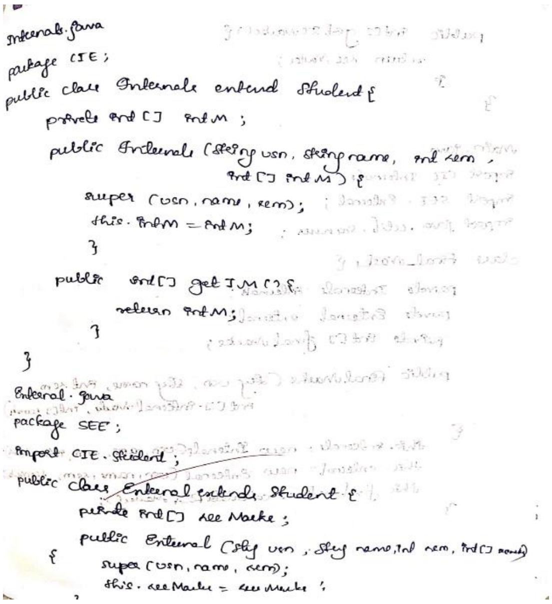
```
4. Compute and Deposit Interest to Savings Account
5. Deposit to Current Account
6. Withdraw from Current Account
7. Display Current Account Balance
8. Exit
Enter your choice: 6
Enter amount to withdraw from Current Account: 4500
Withdrawal successful. Updated balance: 500.0
Balance fell below minimum. Penalty of 100.0 imposed. Updated balance: 400.0

--- Bank Menu ---
1. Deposit to Savings Account
2. Withdraw from Savings Account
3. Display Savings Account Balance
4. Compute and Deposit Interest to Savings Account
5. Deposit to Current Account
6. Withdraw from Current Account
7. Display Current Account Balance
8. Exit
Enter your choice: 8
Exiting program. Thank you!
```

PROGRAM 6

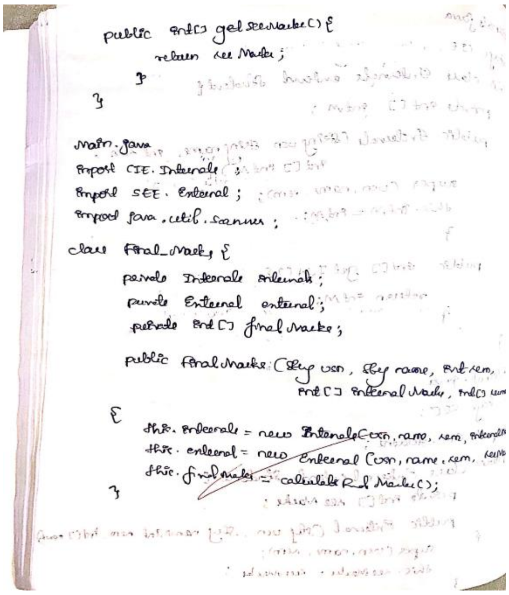
Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.



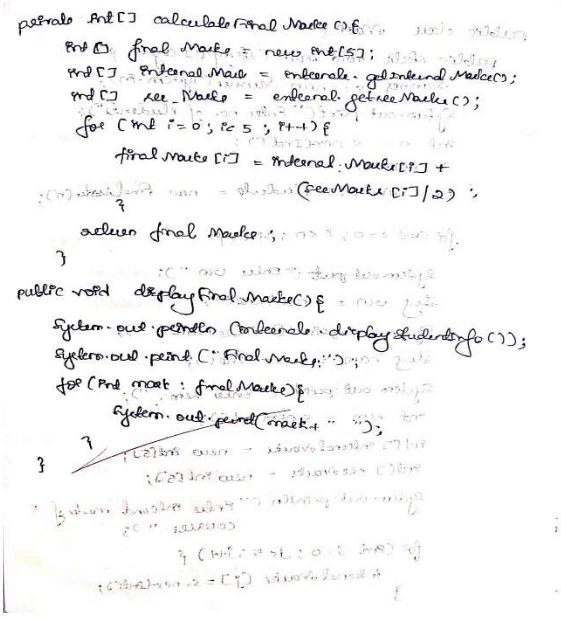
```
Lab-5.
Program:
Student.java
package CIE;
public class Student{
        private String usn;
         private String name;
         private int sem;
    public Student (String usn, String name, int sem){
            this.usn = usn;
            this.name = name;
            this.sem = sem;
        }
    public String displayStudentInfo () {
        return  "usn " + usn + "name : "+name + " sem:"
        }
}
```



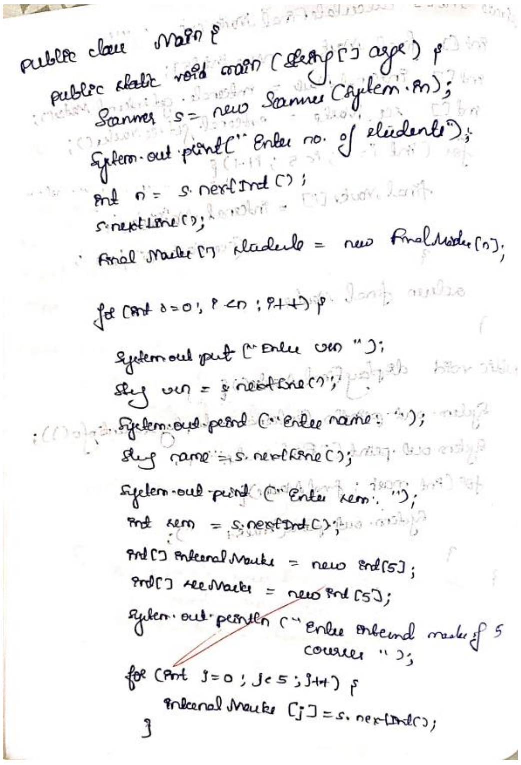
```
Internals.java
package CIE;
public class Internals extend Student {
    private int [] intM ;
    public Internals (String usn, String name, int sem ,
                                int [] intM) {
        super (usn, name, sem);
        this. intM = intM;
        }
    public  int[] get IM () {
            return intM;
        }
}
External . java
package SEE;
import CIE . Student ;
public class External extends Student {
        private int[] see Marks ;
        public External (String usn , String name, int sem, int[] seeMarks)
    {
        super (usn, name, sem);
        this . seeMarks = seeMarks ;
```



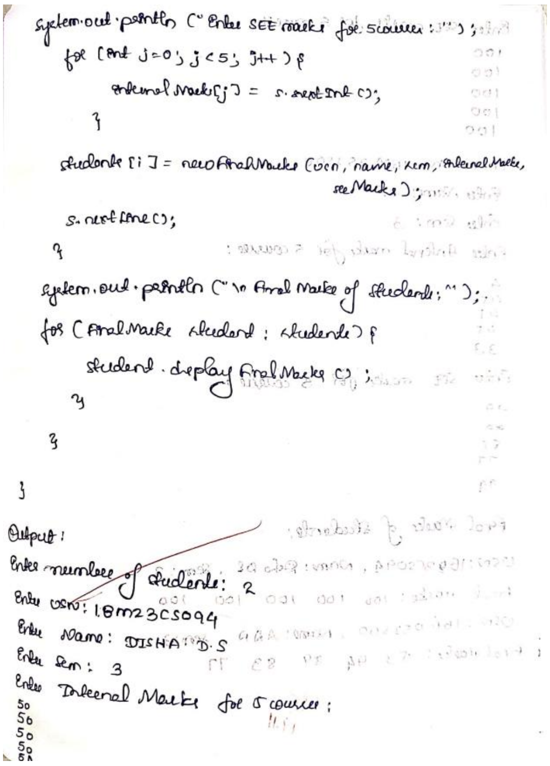
```
    public  int[] getseeMarks() {
            return see Marks ;
        }
  }

Main.java
import CIE. Internals ;
import SEE . External ;
import java . util . Scanner ;
class Final_Marks {
        private Internals internals;
        private External external;
        private int [] final Marks ;

        public FinalMarks (String usn , String name, int sem,
                                int [] internalMarks , int[] seeM
    {
        this. internals = new Internals(usn, name, sem, internalM
        this . external = new External (usn, name, sem, seeM
        this. finalMarks = calculateFinalMarks();
    }
```



```
private int[]  calculateFinalMarks () {
    int []  final Marks = new int[5];
    int []  internal Marks = internals . getInternalMarks();
    int []  see_Marks  =  external. getseeMarks ();
    for (int i = 0; i< 5 ; i++) {
        final Marks [i]  = internal. Marks[i] +
                                    (seeMarks [i]/2) ;
    }
    return final Marks;
}
public void  displayFinalMarks() {
    System . out . println (internals . displayStudentInfo ());
    System.out . print (" Final Marks:");
    for (int mark : final Marks) {
        System. out . print(mark + "  ");
    }
}
```



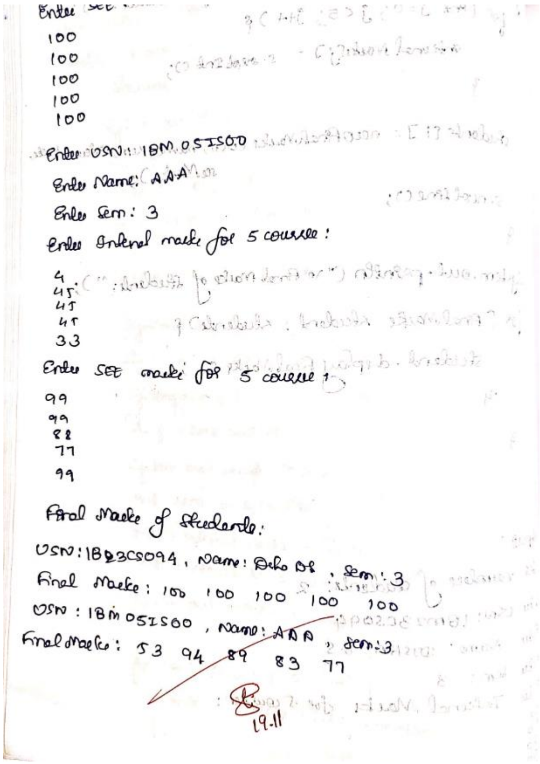
```
public class Main {
    public static void main (String[] args) {
        Scanner s = new Scanner (System.in);
        System.out.print(" Enter no. of students");
        int n = s.nextInt ();
        s.nextLine();
        FinalMarks[] students = new FinalMarks[n];

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

          System.out.print (" Enter usn ");
          String usn = s.nextLine();
          System.out.print (" Enter name: ");
          String name = s.nextLine();
          System.out.print (" Enter sem: ");
          int sem = s.nextInt();
          int[] internalMarks = new int[5];
          int[] seeMarks = new int [5];
          System.out.println (" Enter internal marks of 5
                                 courses ");
          for (int j=0 ; j<5 ; j++) {
              internalMarks [j] = s.nextInt();
          }
```



```
System.out.println (" Enter SEE marks for 5courses: ");
    for (int j=0; j<5; j++) {
          internalMarks[j] = s.nextInt ();
      }

    students [i] = new FinalMarks (usn, name, sem, internalMarks,
                                             seeMarks );

    s.nextLine();
  }
  System.out.println (" \n Final Marks of Students: ");
  for (FinalMarks student : students) {
       student . displayFinalMarks () ;
    }

  }

}
```

Output:

```
Enter number of students: 2
Enter USN: 1BM23CS094
Enter Name: DISHA D.S
Enter Sem: 3
Enter Internal Marks for 5 courses:
50
50
50
50
50
```



```
Enter SEE
 100
 100
 100
 100
 100
Enter USN: 1BM 05IS00
 Enter Name: AAA
 Enter Sem: 3
Enter Internal marks for 5 courses:

 4
 45
 45
 45
 33
Enter SEE marks for 5 courses :
 99
 99
 88
 77
 99

Final Marks of Students:
USN:1BM23CS094, Name: Disha DS , Sem: 3
Final Marks: 100 100 100 100 100
USN : 1BM 05IS00 , Name: AAA , Sem: 3
Final Marks: 53 94 89 83 77
```

19.11

```
package CIE;

public class Student{
   protected String usn;
   protected String name;
   protected int sem;

   public Student(String usn, String name, int sem) {
      this.usn = usn;
      this.name = name;
      this.sem = sem;
   }

   public String displayStudentInfo() {
      return "USN: " + usn + ", Name: " + name + ", Semester: " + sem;
   }
}

package CIE;

public class Internals extends Student {
   private int[] internalMarks; // Array to store internal marks for 5 courses
```

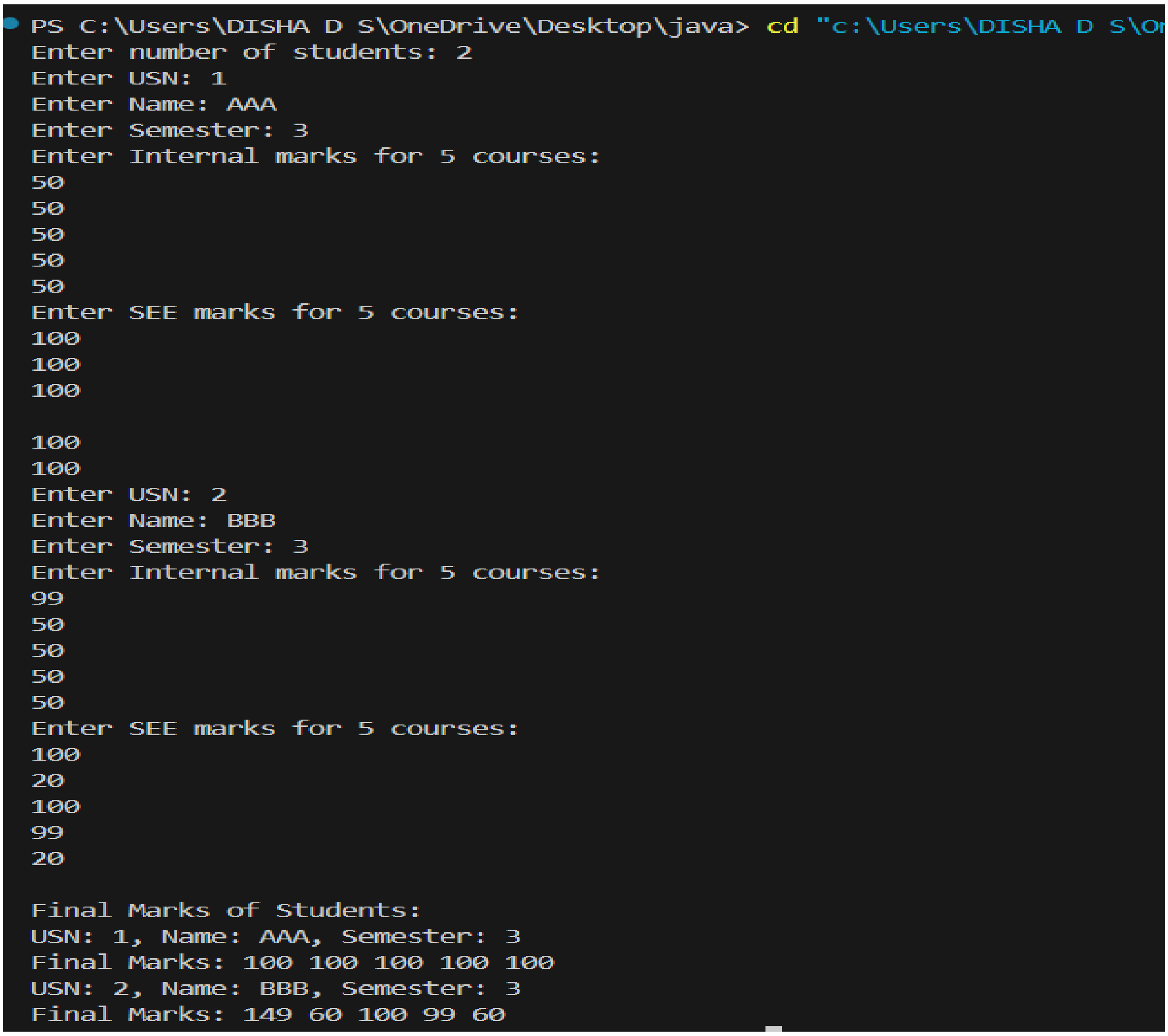
```
    public Internals(String usn, String name, int sem, int[]
internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }

    public int[] getInternalMarks() {
        return internalMarks;
    }
}

package SEE;

import CIE.Student;

public class External extends Student {
    private int[] seeMarks; // Array to store SEE marks for 5 courses

    public External(String usn, String name, int sem, int[] seeMarks) {
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }

    public int[] getSeeMarks() {
        return seeMarks;
    }
}

import CIE.Internals;
import SEE.External;
import java.util.Scanner;

class FinalMarks {
    private Internals internals;
    private External external;
    private int[] finalMarks;

    public FinalMarks(String usn, String name, int sem, int[] internalMarks, int[] seeMarks) {
        this.internals = new Internals(usn, name, sem, internalMarks);
        this.external = new External(usn, name, sem, seeMarks);
        this.finalMarks = calculateFinalMarks();
    }

    private int[] calculateFinalMarks() {
        int[] finalMarks = new int[5];
        int[] internalMarks = internals.getInternalMarks();
        int[] seeMarks = external.getSeeMarks();
        for (int i = 0; i < 5; i++) {
```

```
      // Assuming final mark is (internal mark) + (SEE mark scaled
to 50%)
      finalMarks[i] = internalMarks[i] + (seeMarks[i] / 2);
    }
    return finalMarks;
  }

  public void displayFinalMarks() {
    System.out.println(internals.displayStudentInfo());
    System.out.print("Final Marks: ");
    for (int mark : finalMarks) {
      System.out.print(mark + " ");
    }
    System.out.println();
  }
}

public class Main1 {
  public static void main(String[] args) {
    Scanner s = new Scanner(System.in);

    System.out.print("Enter number of students: ");
    int n = s.nextInt();
    s.nextLine();
```

```
        FinalMarks[] students = new FinalMarks[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter USN: ");
            String usn = s.nextLine();
            System.out.print("Enter Name: ");
            String name = s.nextLine();
            System.out.print("Enter Semester: ");
            int sem = s.nextInt();

            int[] internalMarks = new int[5];
            int[] seeMarks = new int[5];

            System.out.println("Enter Internal marks for 5 courses:");
            for (int j = 0; j < 5; j++) {
                internalMarks[j] = s.nextInt();
            }

            System.out.println("Enter SEE marks for 5 courses:");
            for (int j = 0; j < 5; j++) {
                seeMarks[j] = s.nextInt();
            }

            students[i] = new FinalMarks(usn, name, sem, internalMarks,
seeMarks);
```

```
            s.nextLine();  // Consume newline
        }

        System.out.println("\nFinal Marks of Students:");
        for (FinalMarks student : students) {
            student.displayFinalMarks();
        }
    }
}
```

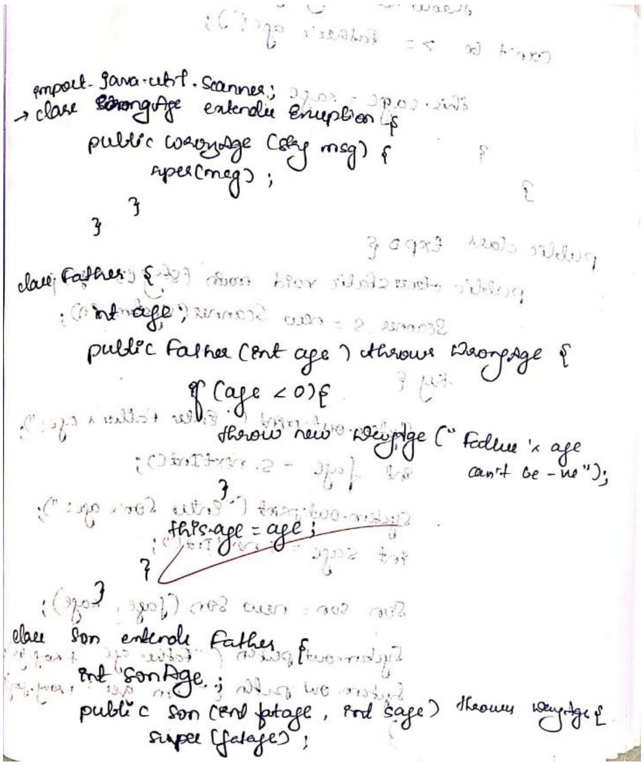
OUTPUT



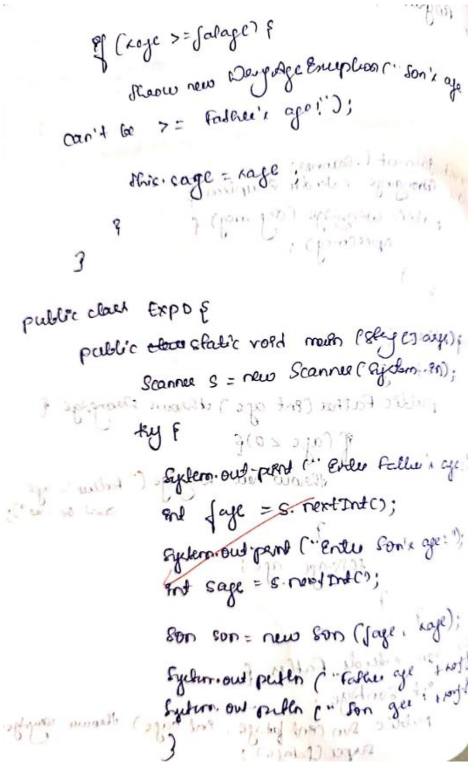
```
PS C:\Users\DISHA D S\OneDrive\Desktop\java> cd "c:\Users\DISHA D S\O
 Enter number of students: 2
 Enter USN: 1
 Enter Name: AAA
 Enter Semester: 3
 Enter Internal marks for 5 courses:
 50
 50
 50
 50
 50
 Enter SEE marks for 5 courses:
 100
 100
 100

 100
 100
 Enter USN: 2
 Enter Name: BBB
 Enter Semester: 3
 Enter Internal marks for 5 courses:
 99
 50
 50
 50
 50
 Enter SEE marks for 5 courses:
 100
 20
 100
 99
 20

 Final Marks of Students:
 USN: 1, Name: AAA, Semester: 3
 Final Marks: 100 100 100 100 100
 USN: 2, Name: BBB, Semester: 3
 Final Marks: 149 60 100 99 60
```

PROGRAM 7

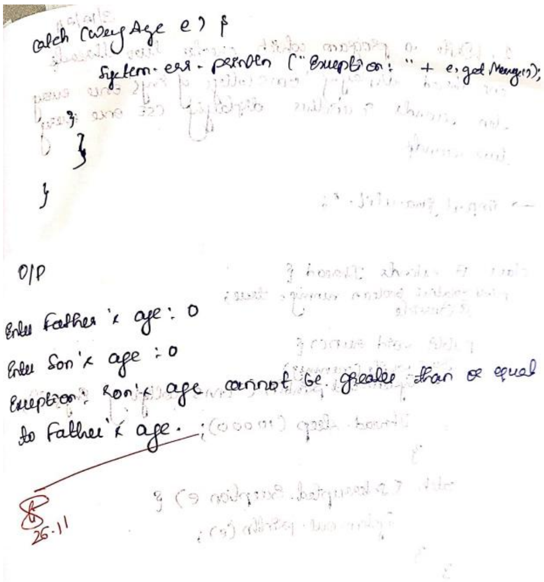
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age=father’s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



```
import java.util.Scanner;
→ class WrongAge extends Exception {
        public WrongAge (String msg) {
                super(msg) ;
        }
    }

class Father {
    int age;
        public Father (int age ) throws WrongAge {
                if (age < 0){
                        throw new WrongAge (" Father 's age
                                                can't be -ve");
                }
                this.age = age ;
        }
    }
class Son extends Father {
    int sonAge ;
    public Son (int fatage , int sage) throws WrongAge {
            super (fatage) ;
```



```
        if (sage >= fatage) {
            throw new WrongAgeException (" Son's age
    can't be >= Father's age!");

            this.sage = sage ;

        }
    }

public class ExpD {
    public ~~class~~ static void main (String [] args);
            Scanner s = new Scanner (System.in);
            try {
                System.out.print (" Enter Father's age
                int fage = s.nextInt();
                System.out.print ("Enter Son's age: ");
                int sage = s.nextInt();

                Son son = new Son (fage, sage);
                System.out.println (" Father age " + fa
                System.out.println (" Son age : " + son
            }
```



```
catch (WrongAge e) {
        System.err.println ("Exception: " + e.getMessage());
    }
  }
}
```

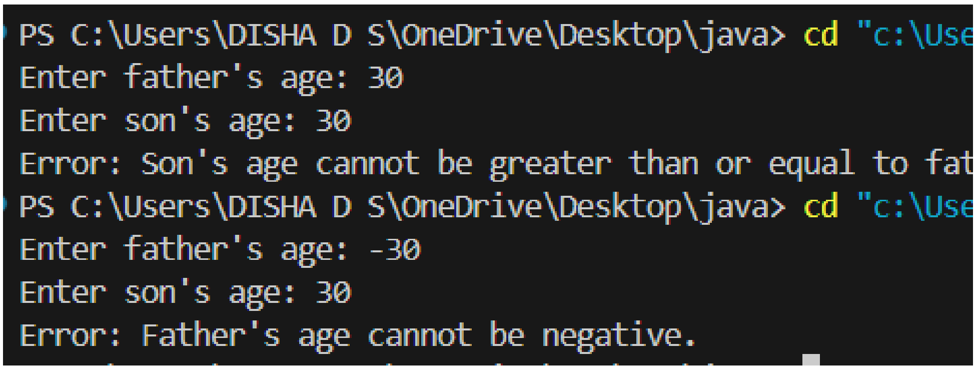
O/P

```
Enter Father's age : 0
Enter Son's age : 0
Exception: Son's age cannot be greater than or equal
to Father's age.
```

26.11

```
import java.util.Scanner;

class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

class Father {
    protected int age;

    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Father's age cannot be negative.");
        }
        this.age = age;
    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be negative.");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to father's age.");
        }
        this.sonAge = sonAge;
    }

    public void displayAges() {
        System.out.println("Father's Age: " + age);
        System.out.println("Son's Age: " + sonAge);
    }
}

public class FamilyAgeTest {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.print("Enter father's age: ");
            int fatherAge = scanner.nextInt();
            System.out.print("Enter son's age: ");
            int sonAge = scanner.nextInt();

            Son son = new Son(fatherAge, sonAge);
            son.displayAges();
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        } catch (Exception e) {
            System.out.println("Invalid input. Please enter valid
ages.");
        }
    }}
```

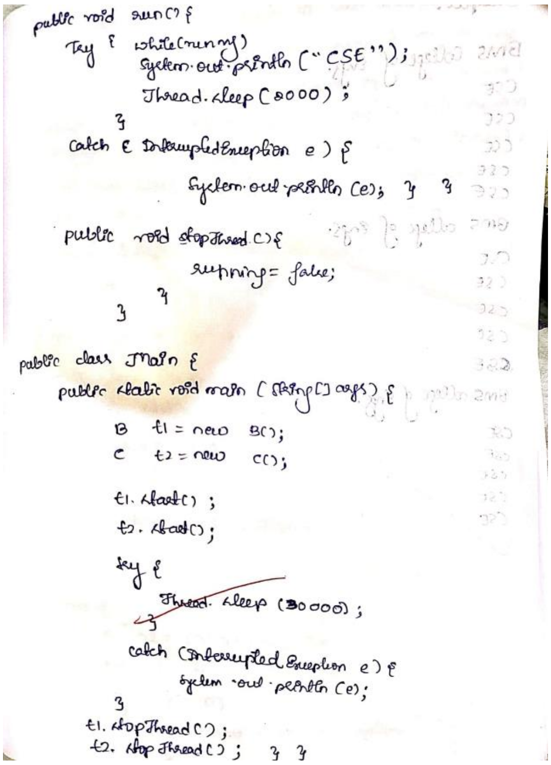
Output:



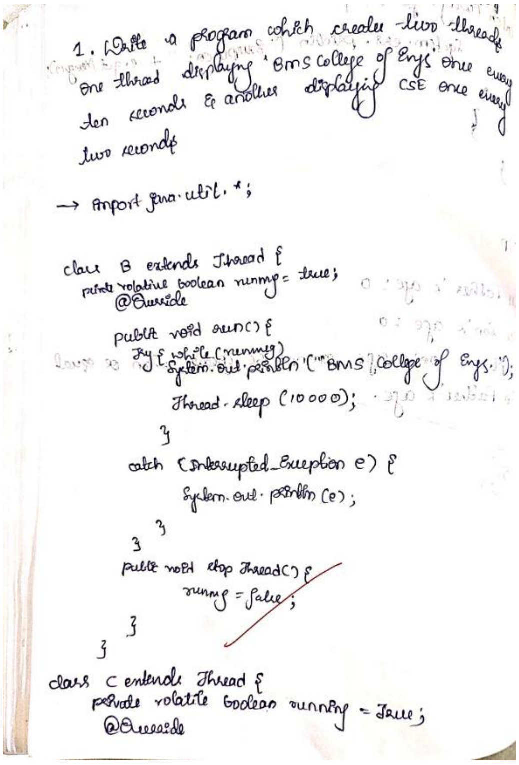
```
PS C:\Users\DISHA D S\OneDrive\Desktop\java> cd "c:\Use
Enter father's age: 30
Enter son's age: 30
Error: Son's age cannot be greater than or equal to fat
PS C:\Users\DISHA D S\OneDrive\Desktop\java> cd "c:\Use
Enter father's age: -30
Enter son's age: 30
Error: Father's age cannot be negative.
```

PROGRAM 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.

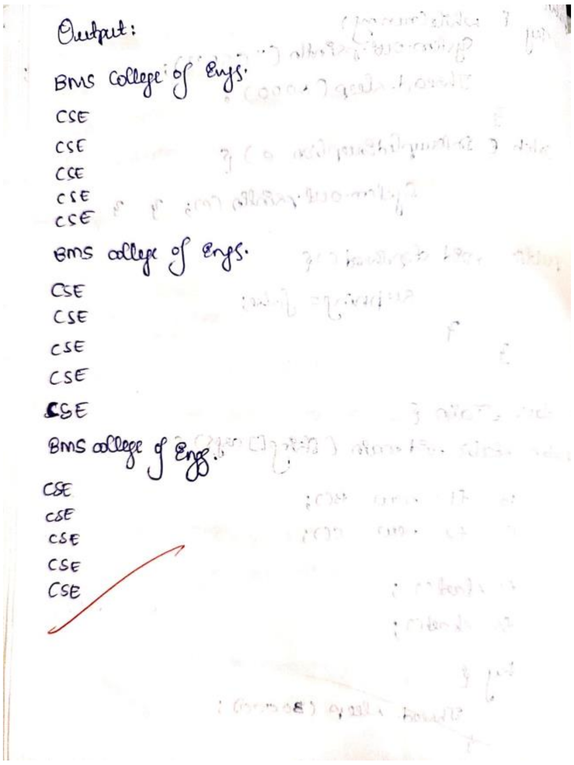


```
public void run() {
    Try {  while(running)
           System.out.println ("CSE");
           Thread.sleep (2000) ;
        }
    catch ( InterruptedException e ) {
                System.out.println (e);  }   }

    public  void stopThread () {
                running= false;
            }
        }

public class Main {
    public static void main ( String[] args) {
            B   t1 = new   B();
            C   t2 = new   C();

            t1. start() ;
            t2. start();

            try {
                Thread. sleep (30000) ;
            }
            catch (InterruptedException e) {
                    System .out .println (e);
            }
        t1. stopThread () ;
        t2. stopThread () ;    }  }
```



1. Write a program which creates two threads one thread displaying 'BMS College of Engg' once every ten seconds & another displaying CSE once every two seconds

```
→ import java.util.*;

class B extends Thread {
  private volatile boolean running= true;
      @Override
      public void run() {
        Try { while(running)
              System.out.println ("BMS College of Engg.");
              Thread.sleep (10000);
          }
        catch (Interrupted_Exception e) {
              System.out.println (e);
          }
      }
      public void stop Thread() {
              running = false;
      }
  }
class C extends Thread {
    private volatile boolean running = True;
    @Override
```

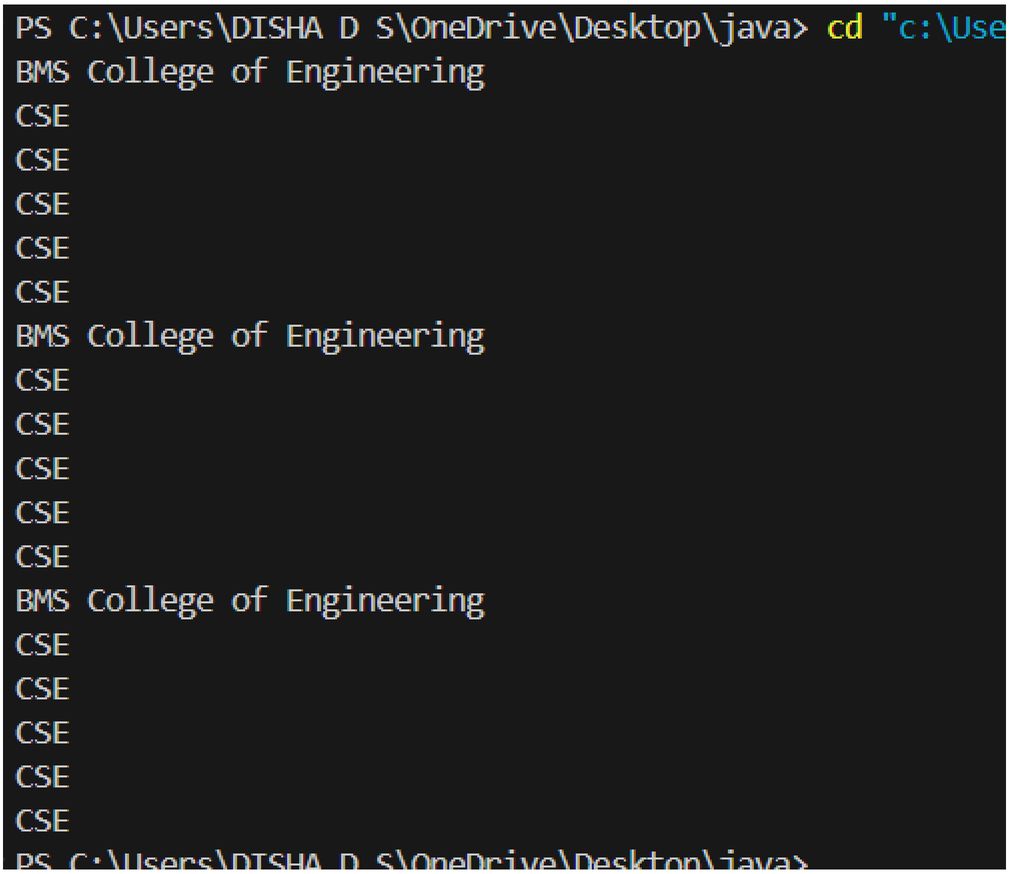


Output:

```
BMS College of Engg.
CSE
CSE
CSE
CSE
CSE
BMS college of Engg.
CSE
CSE
CSE
CSE
CSE
BMS college of Engg.
CSE
CSE
CSE
CSE
CSE
```

```
class B extends Thread {
    private volatile boolean running = true;

    @Override
    public void run() {
        try {
            while (running) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }

    public void stopThread() {
        running = false;
    }
}

class C extends Thread {
    private volatile boolean running = true;

    @Override
    public void run() {
        try {
            while (running) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }

    public void stopThread() {
        running = false;
    }
}

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

        B thread1 = new B();
        C thread2 = new C();
        thread1.start();
        thread2.start();

        try {
            Thread.sleep(30000);
        } catch (InterruptedException e) {
            System.out.println(e);
    }
        thread1.stopThread();
        thread2.stopThread();
    }
}
```

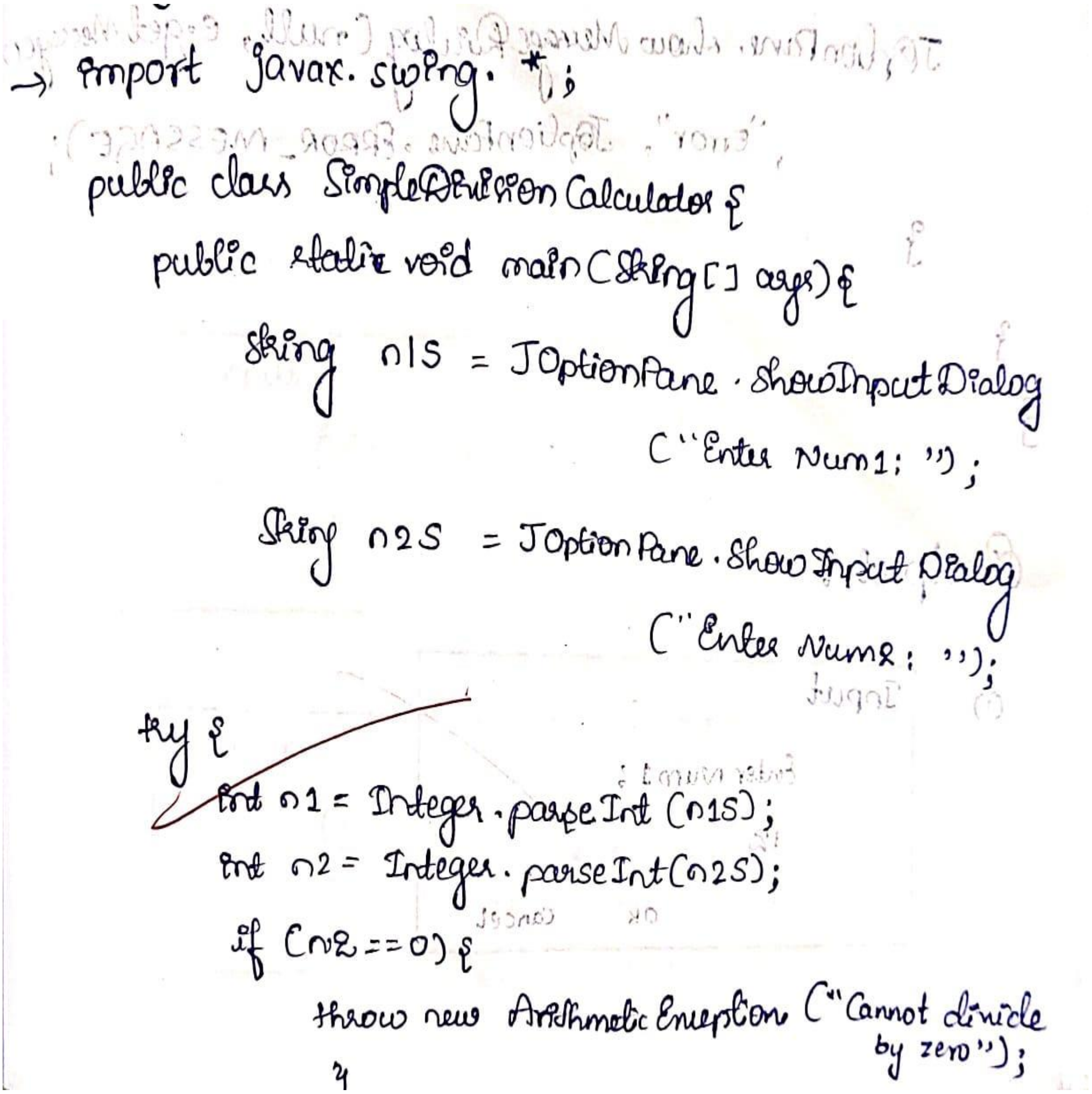
Output



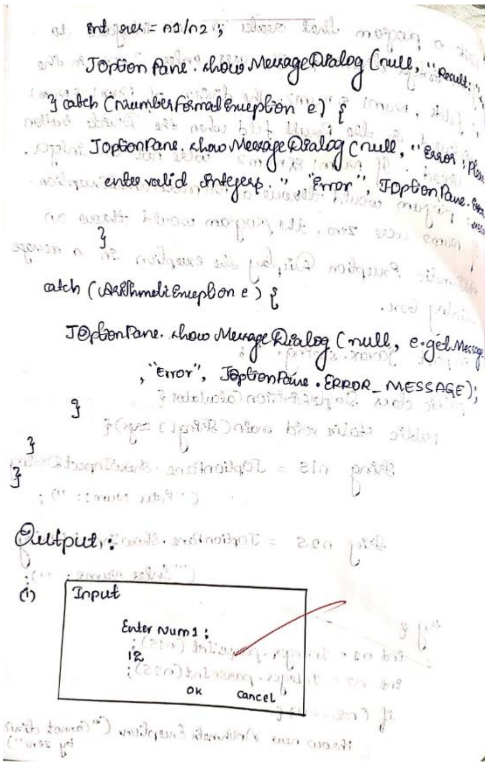
```
PS C:\Users\DISHA D S\OneDrive\Desktop\java> cd "c:\Use
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
PS C:\Users\DISHA D S\OneDrive\Desktop\java>
```

Program 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 javax.swing.*;

  public class SimpleDivision Calculator {
     public static void main(String[] args){
           String  n1S  = JOptionPane.showInputDialog
                                     ("Enter Num1: ");
           String  n2S  = JOptionPane.showInputDialog
                                     ("Enter Num2: ");
     try {
         int n1 = Integer.parseInt (n1S);
         int n2 = Integer.parseInt(n2S);
         if (n2==0) {
              throw new ArithmeticException ("Cannot divide
                                                  by zero");
         }
```

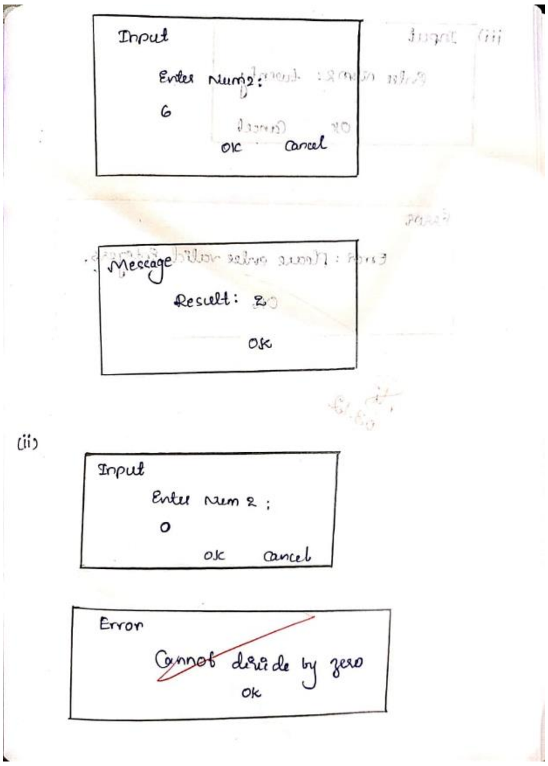


```
    int res = n1/n2 ;
        JOption Pane . show MessageDialog (null, "Result:
    } catch (NumberFormatException e) {
        JOptionPane. show MessageDialog (null, "Error ; Plea
          enter valid integers. ", "Error", JOptionPane.
     }

  catch (ArithmeticException e) {

    JOptionPane. show MessageDialog (null, e.getMessag
                , "Error", JOptionPane . ERROR_MESSAGE);
   }
  }
}
```

Output :

(i)

```
Input
      Enter Num1 :
      12
              OK      Cancel
```



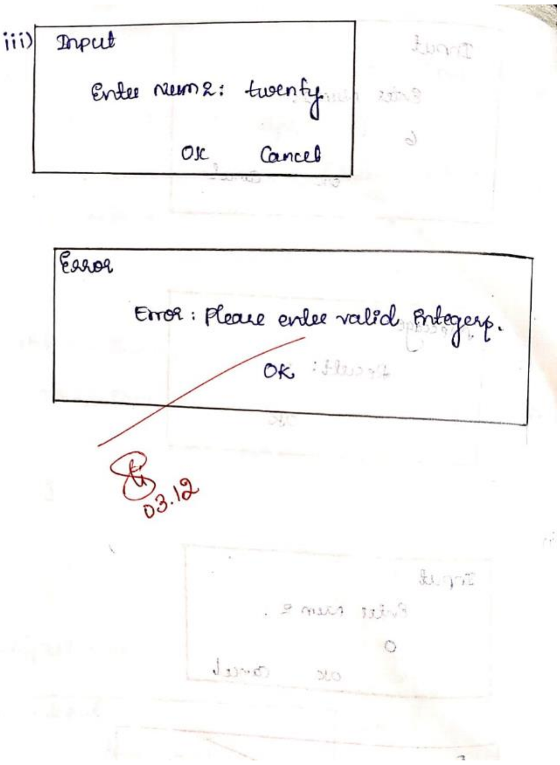
```
Input
      Enter Num2 :
      6
              OK      Cancel
```

```
Message
      Result: 2
              OK
```

(ii)

```
Input
      Enter Num 2 :
      0
              OK      Cancel
```

```
Error
      Cannot divide by zero
              OK
```



iii)

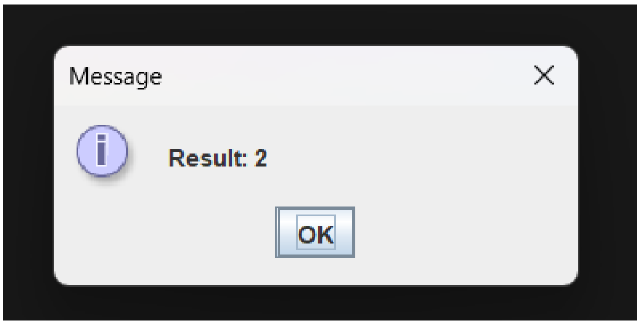
```
Input
      Enter Num2: twenty
              OK      Cancel
```

```
Error
      Error : Please enter valid integers.
              OK
```

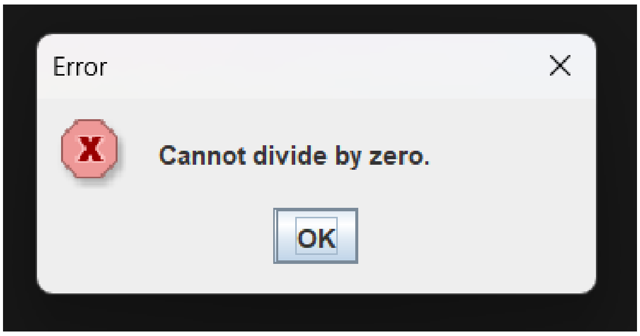
03.12

```
import javax.swing.*;

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

        String num1Str = JOptionPane.showInputDialog("Enter Num1:");
        String num2Str = JOptionPane.showInputDialog("Enter Num2:");

        try {

            int num1 = Integer.parseInt(num1Str);
            int num2 = Integer.parseInt(num2Str);



            if (num2 == 0) {

                throw new ArithmeticException("Cannot divide by zero.");
            }

            int result = num1 / num2;
            JOptionPane.showMessageDialog(null, "Result: " + result);
            } catch (NumberFormatException e) {

                JOptionPane.showMessageDialog(null, "Error: Please
enter valid integers.", "Error", JOptionPane.ERROR_MESSAGE);
            } catch (ArithmeticException e) {

                JOptionPane.showMessageDialog(null, e.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }
```

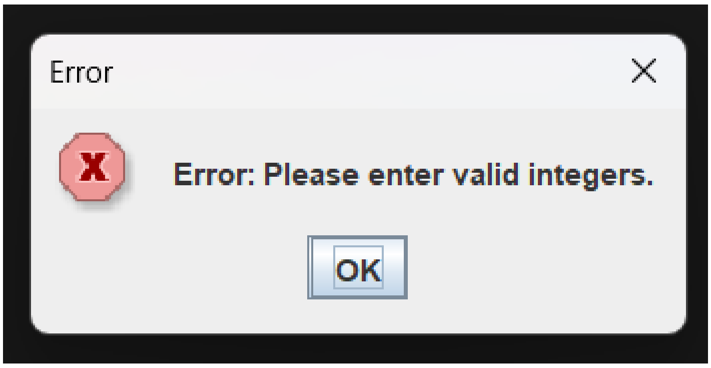
Output:



Result: 2



Cannot divide by zero.



Error: Please enter valid integers.

Github link:

https://github.com/DishaDS094/DISHADS_JAVA_094