VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Hiran B (1BM23CS113)

in partial fulfilment 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 to Jan-2025

B.M.S. College of Engineering,
Bull Temple Road, Bengaluru 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 (23CS3PCOOJ)” carried out by **Hiran B(1BM23CS113),** 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.

Ambuja Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

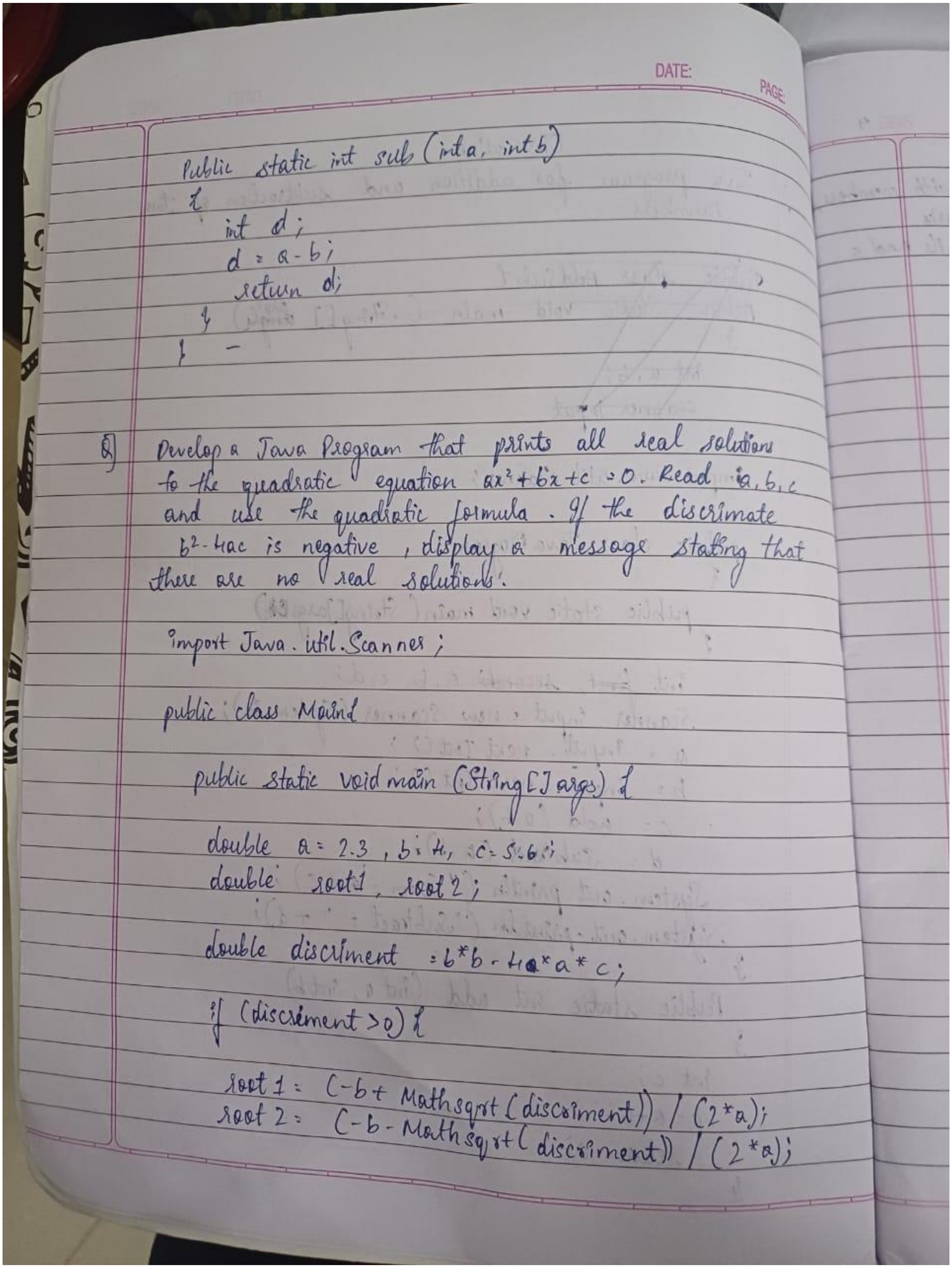
Sl. No.	Date	Experiment Title	Page No.
1	1/10/24	Roots of Quadratic Equations	4-9
2	8/10/24	SGPA Calculator	10-17
3	15/10/24	Method Overriding	18-25
4	22/10/24	Abstract Class	25-31
5	29/10/24	Bank Account	31-41
6	19/11/24	Packages	41-49
7	26/11/24	Exception handling	49-53
8	3/12/24	Threads	53-56
9	3/12/24	Calculator	56-62

GitHub Link:
https://github.com/Hiran20/OOJ_LAB_1BM23CS113

Program 1

Implement Quadratic Equation

Algorithm:



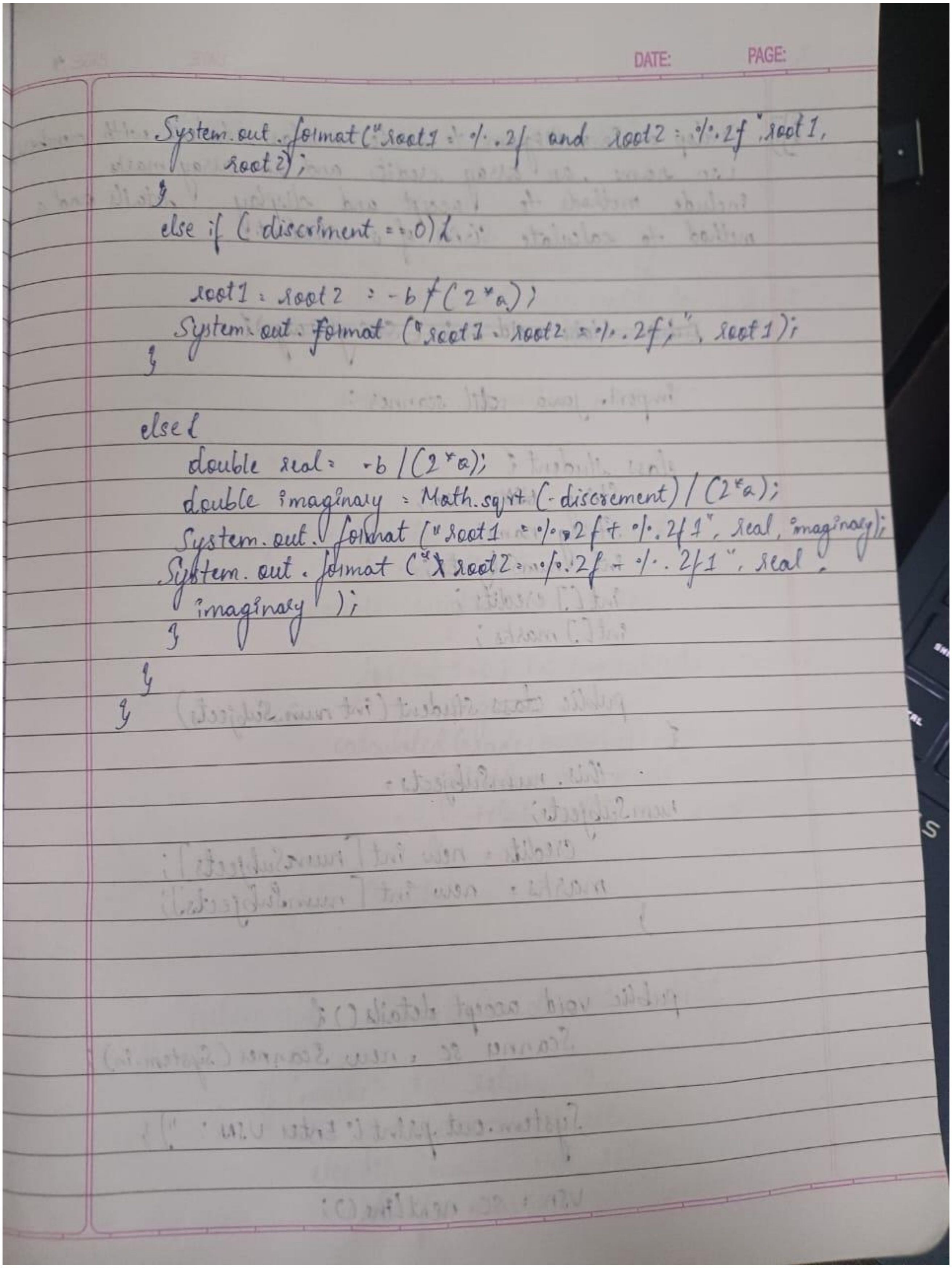
DATE:

PAGE:

```
Public static int sub (int a, int b)
{
    int d;
    d = a-b;
    return d;
  }
}  -
```

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

```
import Java.util.Scanner;

public class Main{

  public static void main (String[] args) {

    double a = 2.3, b = 4, c = 5.6;
    double root1, root2;

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

    if (discriment >0) {

      root 1 = (-b + Mathsqrt (discriment)) / (2*a);
      root 2 = (-b - Mathsqrt (discriment)) / (2*a);
```



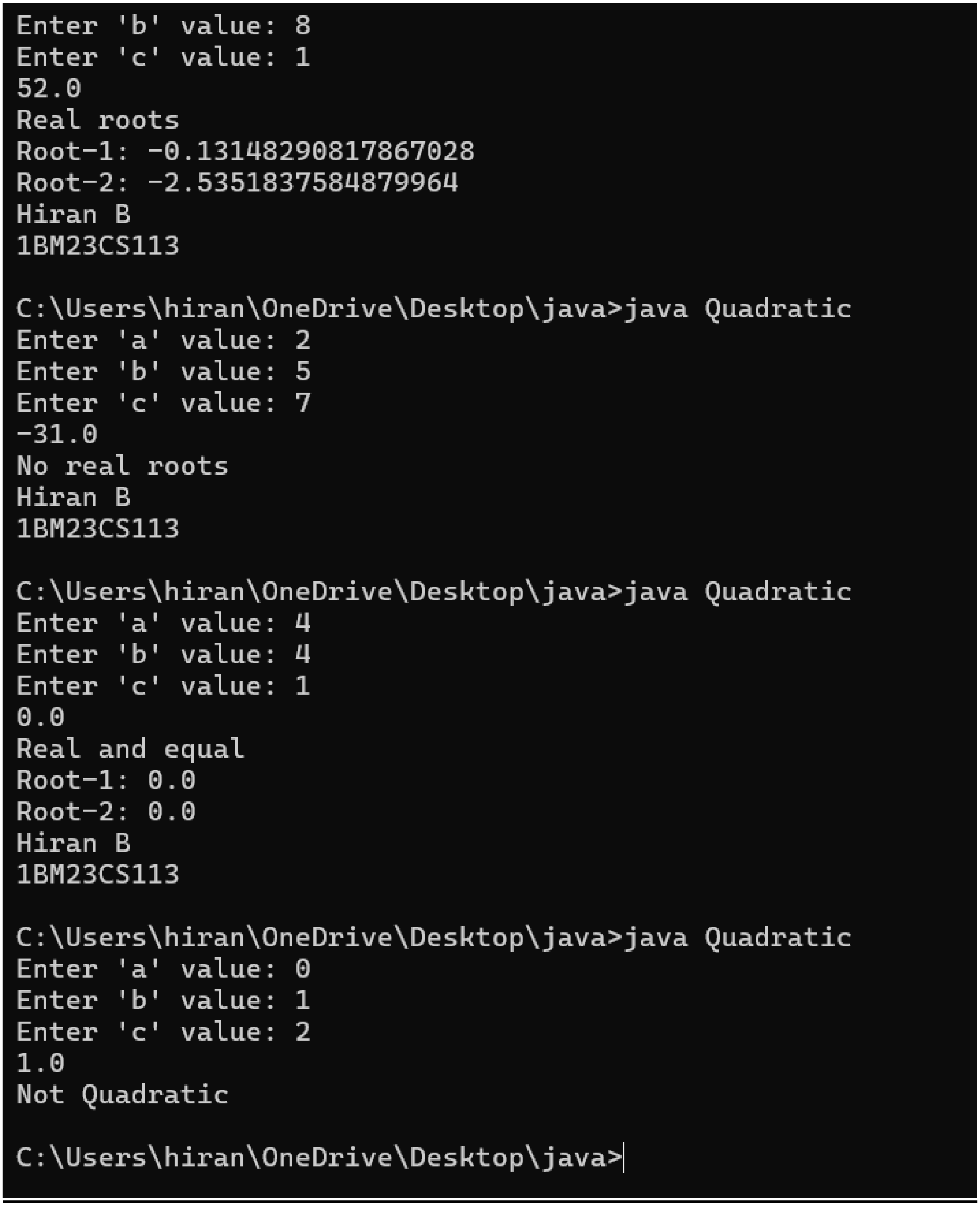
DATE: PAGE:

```
System.out.format("root1 = %.2f and root2 = %.2f",root1,
        root2);
}.
else if ( discriment == 0){

    root1 = root2 = -b/(2*a);
   System.out.format ("root1 = root2 = %.2f;", root1);
}

else{
    double real = -b/(2*a);
    double imaginary = Math.sqrt(-discrement)/(2*a);
    System.out.format ("root1 = %.2f + %.2fi", real, imaginary);
   System.out.format ("x root2 = %.2f - %.2fi", real,
     imaginary );
   }
 }
}
```

Code:

```
import java.util.Scanner;
public class Quadratic

{
    public static void main(String[] args)
    {
        int a;
        int b;
        int c;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter 'a' value: ");
        a= sc.nextInt();
        System.out.print("Enter 'b' value: ");
        b=sc.nextInt();
        System.out.print("Enter 'c' value: ");
        c=sc.nextInt();
        float disc = ((b*b)-4*a*c);
        System.out.println(disc);
        if (a==0)
        {
            System.out.println("Not Quadratic");
        }
        else
        {
            if (disc<0)
            {
            System.out.println("No real roots ");
            }
            else if (disc>0)
            {
            double root1= (-b + Math.sqrt(disc))/(2*a);
            double root2= (-b - Math.sqrt(disc))/(2*a);
            System.out.println("Real roots ");
            System.out.println("Root-1: "+root1);
            System.out.println("Root-2: "+root2);
            }
```

```
           else
           {
           double root1=(-b)/(2*a);
             System.out.println("Real and equal");
           System.out.println("Root-1: "+root1);
           System.out.println("Root-2: "+root1);
           }
         System.out.println("Hiran B");
         System.out.println("1BM23CS113");

       }
    }
}
```

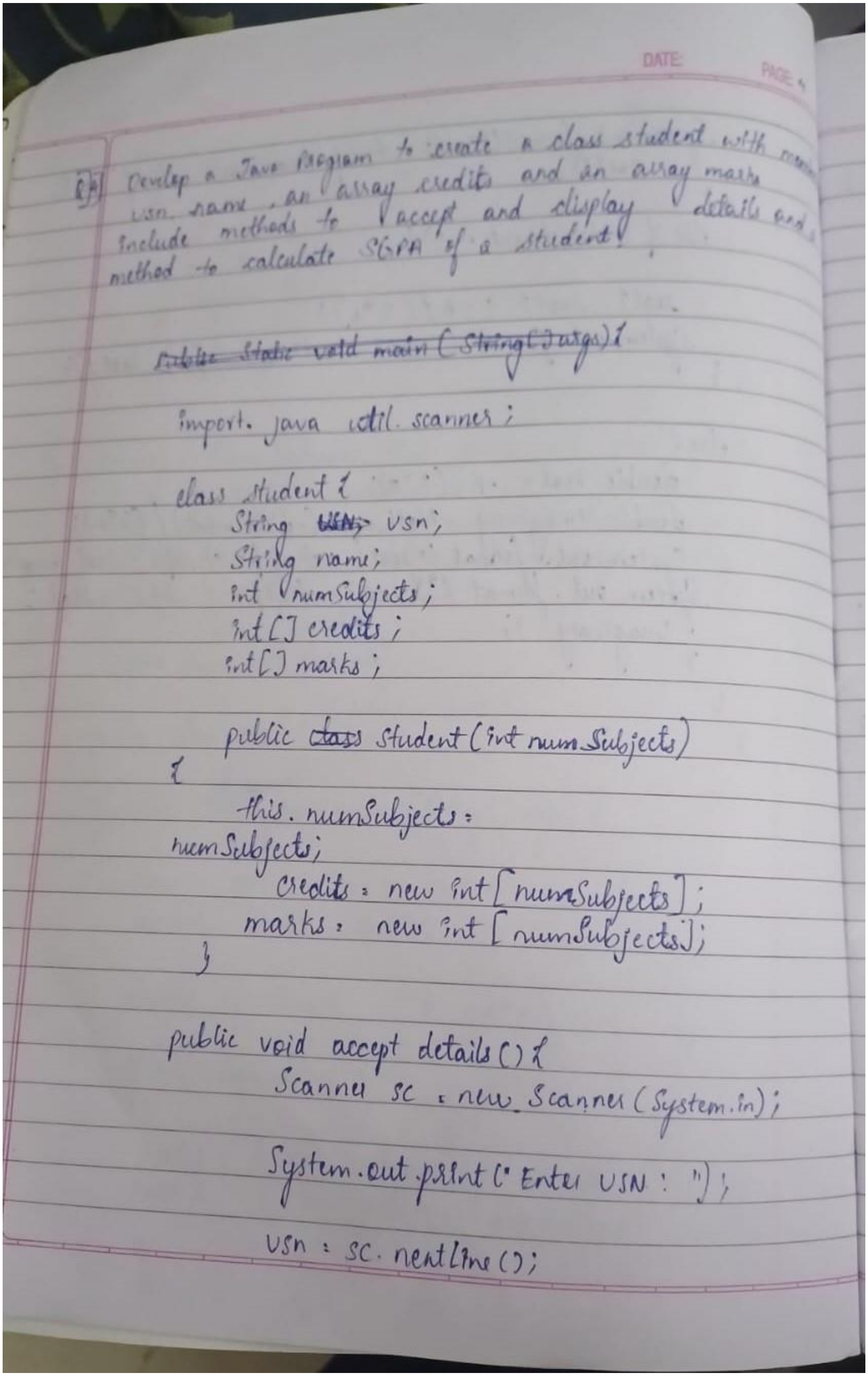


```
Enter 'b' value: 8
Enter 'c' value: 1
52.0
Real roots
Root-1: -0.13148290817867028
Root-2: -2.5351837584879964
Hiran B
1BM23CS113

C:\Users\hiran\OneDrive\Desktop\java>java Quadratic
Enter 'a' value: 2
Enter 'b' value: 5
Enter 'c' value: 7
-31.0
No real roots
Hiran B
1BM23CS113

C:\Users\hiran\OneDrive\Desktop\java>java Quadratic
Enter 'a' value: 4
Enter 'b' value: 4
Enter 'c' value: 1
0.0
Real and equal
Root-1: 0.0
Root-2: 0.0
Hiran B
1BM23CS113

C:\Users\hiran\OneDrive\Desktop\java>java Quadratic
Enter 'a' value: 0
Enter 'b' value: 1
Enter 'c' value: 2
1.0
Not Quadratic

C:\Users\hiran\OneDrive\Desktop\java>
```

Program 2

SGPA Calculator

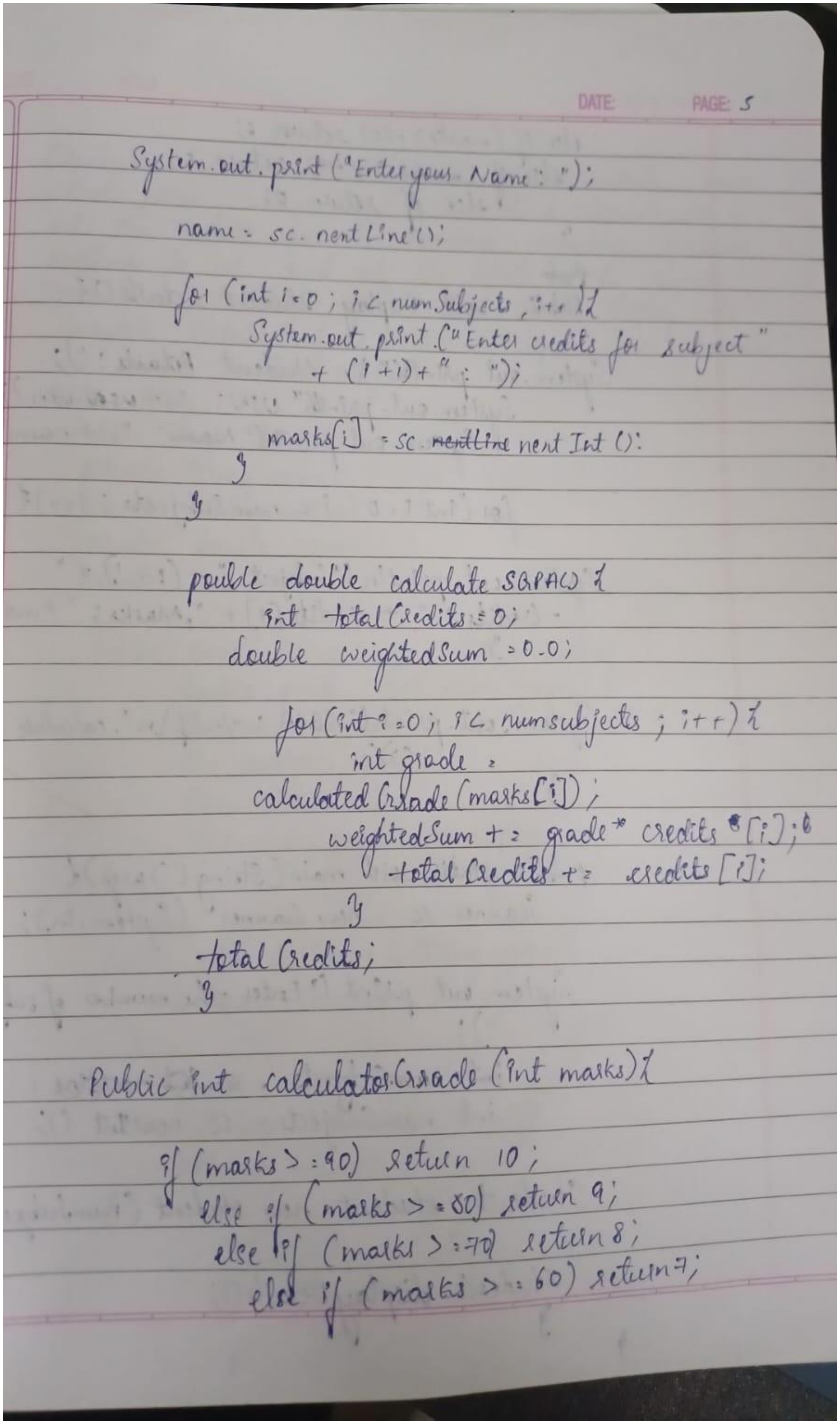
Algorithm:



DATE: PAGE:

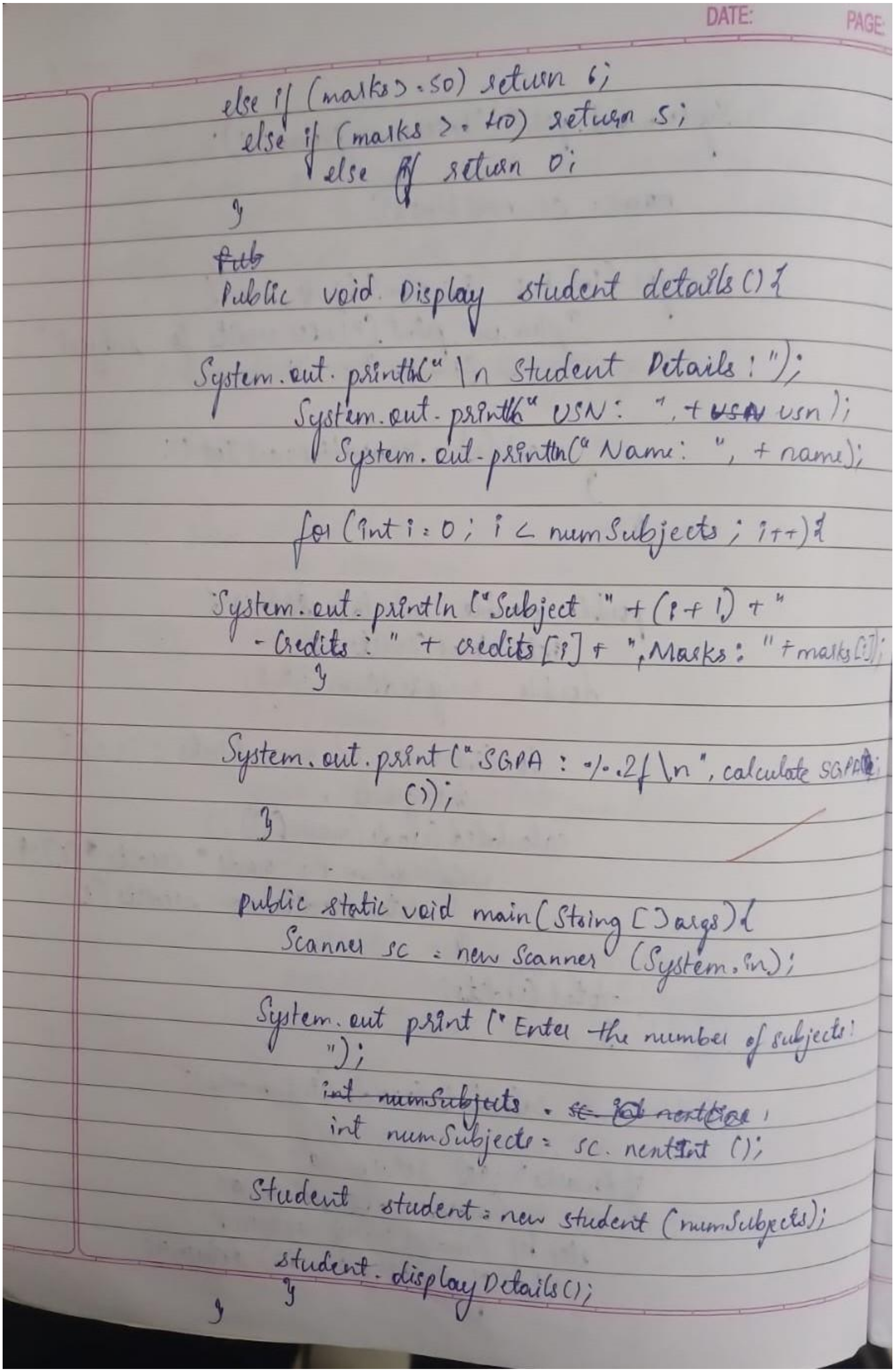
Q.A] Develop a Java Program to create a class student with mem usn, name, an array credits and an array marks Include methods to accept and display details and method to calculate SGPA of a student.

```
~~public Static void main (String[] args){~~

import. java util. scanner;

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

    public ~~class~~ Student (int num Subjects)
{
    this. numSubjects =
numSubjects;
      credits = new int [numSubjects];
      marks = new int [numSubjects];
  }

public void accept details () {
        Scanner sc = new Scanner (System.in);

        System.out.print (" Enter USN : ");

        usn = sc. nextLine ();
```



DATE: PAGE: 5

```
System.out.print ("Enter yous Name : ");

    name = sc. nextLine();

    for (int i=0 ; i< numSubjects , i++ ){
        System.out.print ("Enter credits for subject "
              + (i+1)+ " : ");

        marks[i] = sc. ~~nextLine~~ next Int ():
      }
    }

    pouble double calculate SGPA() {
          int totalCredits = 0;
       double weightedSum =0.0;

           for(int i =0; i< numsubjects ; i++) {
                int grade =
         calculated Grade (marks[i]);
              weightedSum += grade * credits [i];
                 totalCredits += credits [i];
              }
     total Credits;
    }

Public int calculatorGrade (int marks){

    if (marks >=90) return 10;
     else if (marks >=80) return 9;
      else if (marks >=70) return 8;
       else if (marks >= 60) return7;
        else if
```

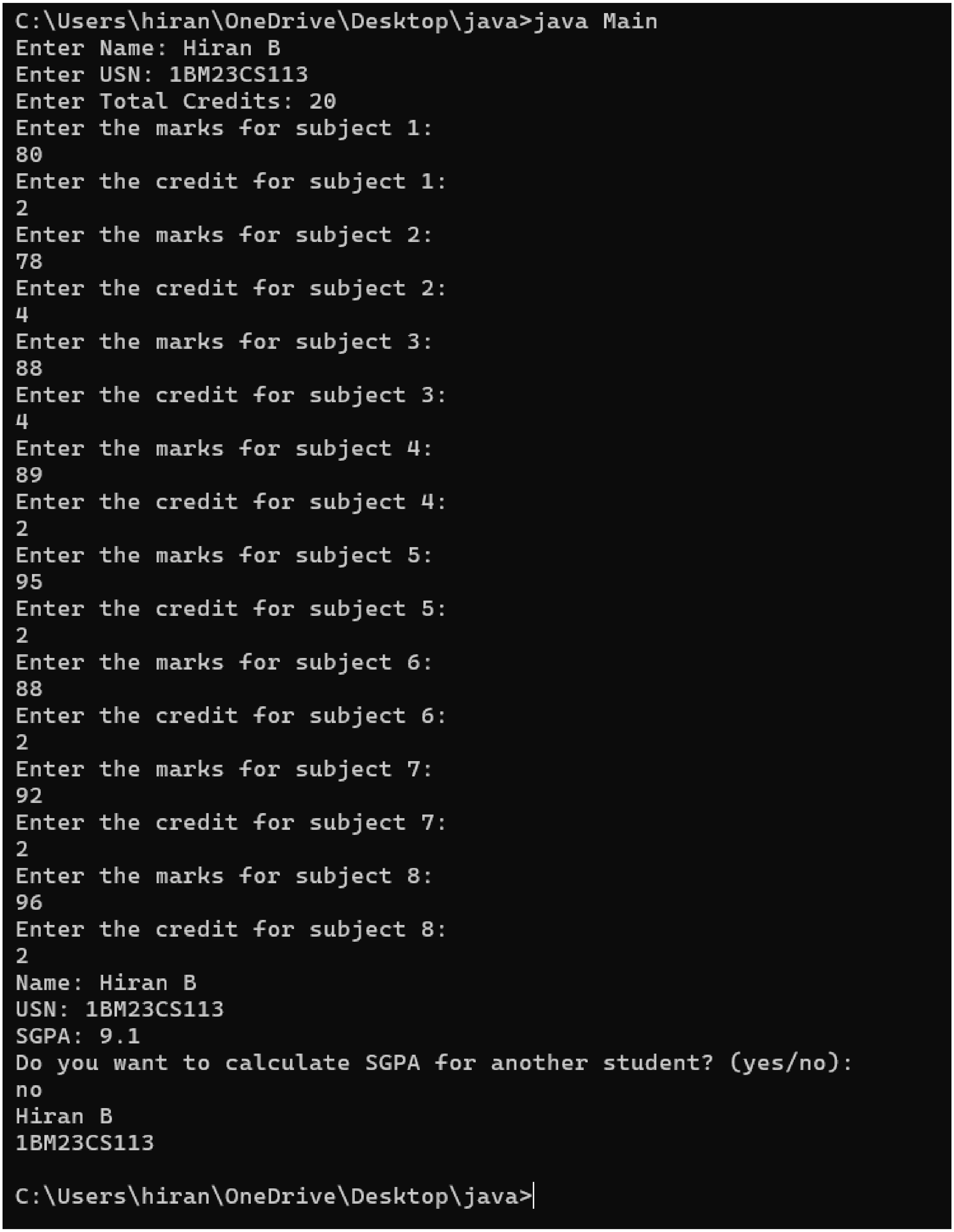


DATE: PAGE:

```
else if (marks >= 50) return 6;
 else if (marks >= 40) return 5;
    else ~~if~~ return 0;
}
~~Pub~~
Public void Display student details(){

System.out.println("\n Student Details:");
    System.out.println(" USN:  ", + ~~USN~~ usn);
     System.out.println(" Name:  ", + name);

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

System.out.println ("Subject " + (i + 1) + "
 - Credits: " + credits[i] + ", Marks: " + marks[i]);
    }

System.out.print ("SGPA : %.2f \n", calculate SGPA
        ());
  }

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

  System.out print (" Enter the number of subjects:
      ");
      ~~int numSubjects = sc. [illegible] next[illegible]~~;
      int numSubjects = sc. nextInt ();

  Student student = new student (numSubjects);

   student. displayDetails();
   }
}
```

Code:

```
import java.util.Scanner;

class Student {
    private String name;
    private String usn;
    private double total_credit;
    private double[] marks;
    private Scanner sc = new Scanner(System.in);

    void getInfo() {
        System.out.print("Enter Name: ");
        name = sc.nextLine();

        System.out.print("Enter USN: ");
        usn = sc.nextLine();

        System.out.print("Enter Total Credits: ");
        total_credit = sc.nextDouble();
        sc.nextLine();
    }

    double grade(double mark) {
        if (mark <= 39) {
            return 0;
        } else if (mark >= 40 && mark <= 49) {
            return 4;
        } else if (mark >= 50 && mark <= 54) {
            return 5;
        } else if (mark >= 55 && mark <= 59) {
            return 6;
        } else if (mark >= 60 && mark <= 69) {
            return 7;
        } else if (mark >= 70 && mark <= 79) {
            return 8;
        } else if (mark >= 80 && mark <= 89) {
            return 9;
        } else {
            return 10;
        }
    }

    void getMarks() {
        marks = new double[8];
        for (int i = 0; i < 8; i++) {
            System.out.println("Enter the marks for subject " + (i + 1) + ": ");
            double mark = sc.nextDouble();

            System.out.println("Enter the credit for subject " + (i + 1) + ": ");
            double credit = sc.nextDouble();

            double grade = grade(mark);
            marks[i] = grade * credit;
        }
        sc.nextLine();
    }

    void calSgpa() {
        double totalMarks = 0;
        for (int i = 0; i < 8; i++) {
            totalMarks += marks[i];
        }
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.println("SGPA: " + (totalMarks / total_credit));
    }
}

public class Main {
    public static void main(String args[]) {
        boolean cond = true;
        Scanner sc = new Scanner(System.in);
        while (cond) {
            Student s1 = new Student();


            s1.getInfo();
            s1.getMarks();
            s1.calSgpa();

            System.out.println("Do you want to calculate SGPA for another student? (yes/no): ");
            String check = sc.nextLine();
            if (check.equalsIgnoreCase("yes")) {
                continue;
            } else {
                cond = false;
            }
        }
          System.out.println("Hiran B");
        System.out.println("1BM23CS113");
        sc.close();
    }
}
```

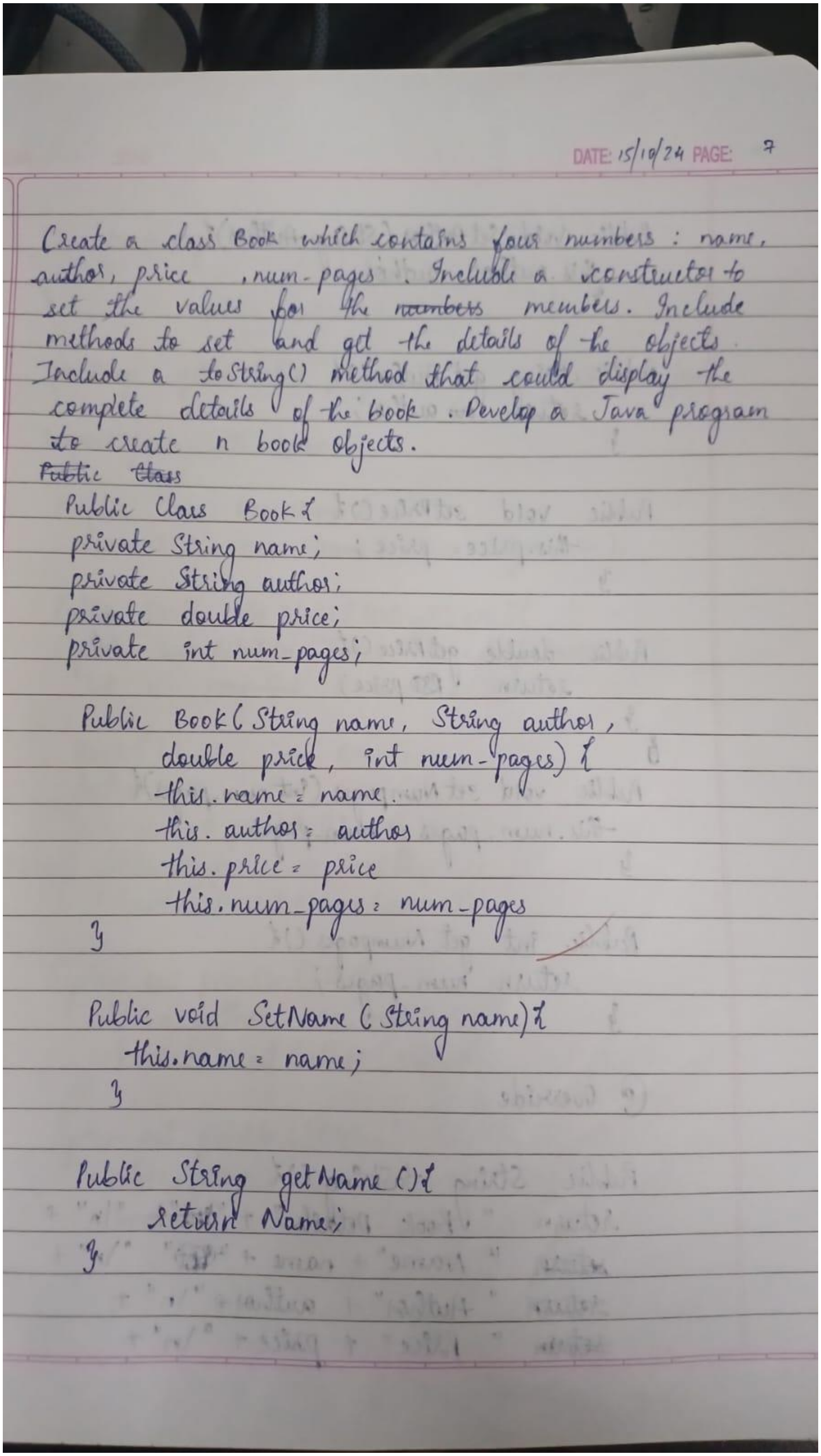


```
C:\Users\hiran\OneDrive\Desktop\java>java Main
Enter Name: Hiran B
Enter USN: 1BM23CS113
Enter Total Credits: 20
Enter the marks for subject 1:
80
Enter the credit for subject 1:
2
Enter the marks for subject 2:
78
Enter the credit for subject 2:
4
Enter the marks for subject 3:
88
Enter the credit for subject 3:
4
Enter the marks for subject 4:
89
Enter the credit for subject 4:
2
Enter the marks for subject 5:
95
Enter the credit for subject 5:
2
Enter the marks for subject 6:
88
Enter the credit for subject 6:
2
Enter the marks for subject 7:
92
Enter the credit for subject 7:
2
Enter the marks for subject 8:
96
Enter the credit for subject 8:
2
Name: Hiran B
USN: 1BM23CS113
SGPA: 9.1
Do you want to calculate SGPA for another student? (yes/no):
no
Hiran B
1BM23CS113

C:\Users\hiran\OneDrive\Desktop\java>
```

Program 3

Method overriding

Algorithm:

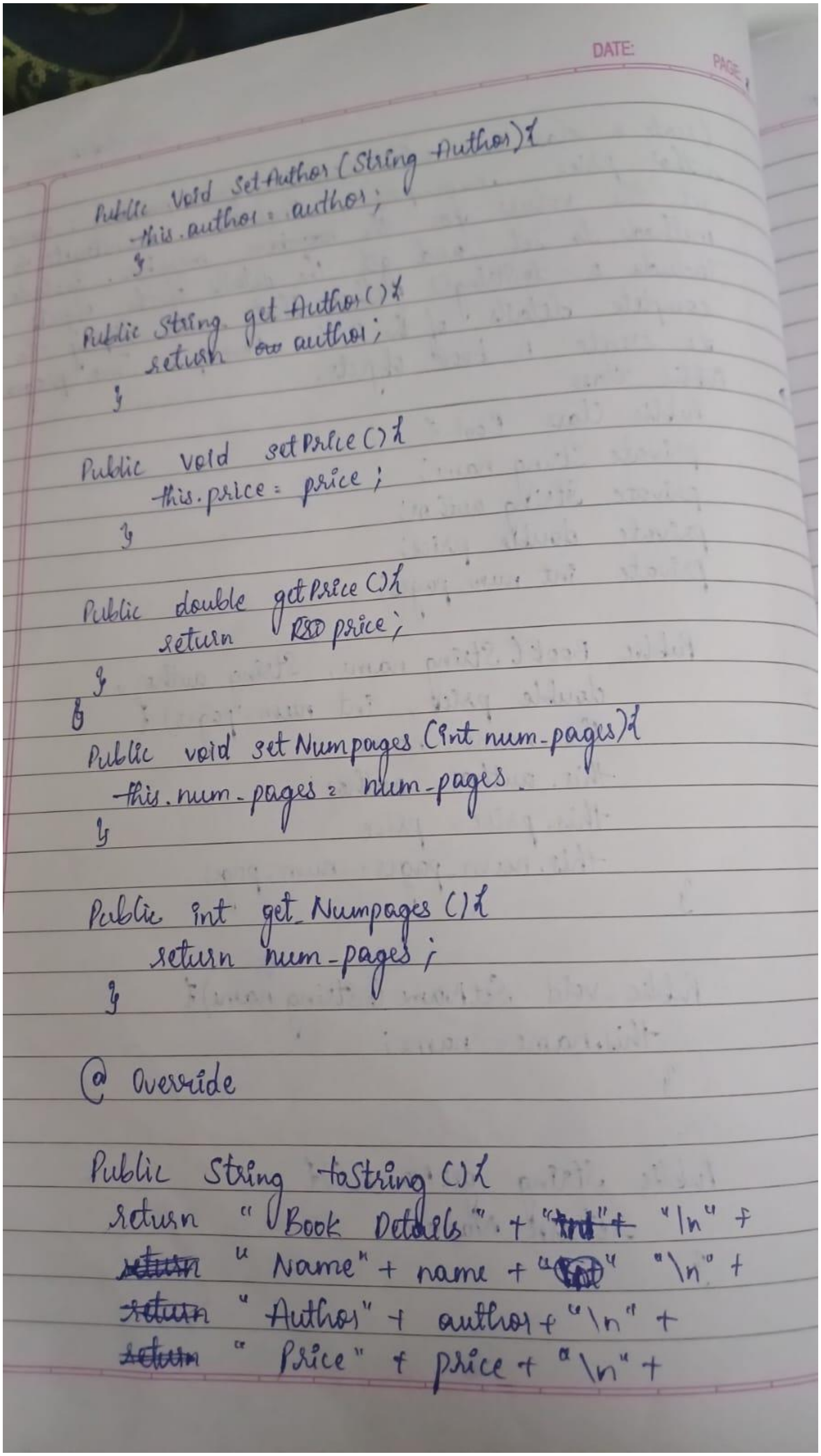


DATE: 15/10/24 PAGE: 7

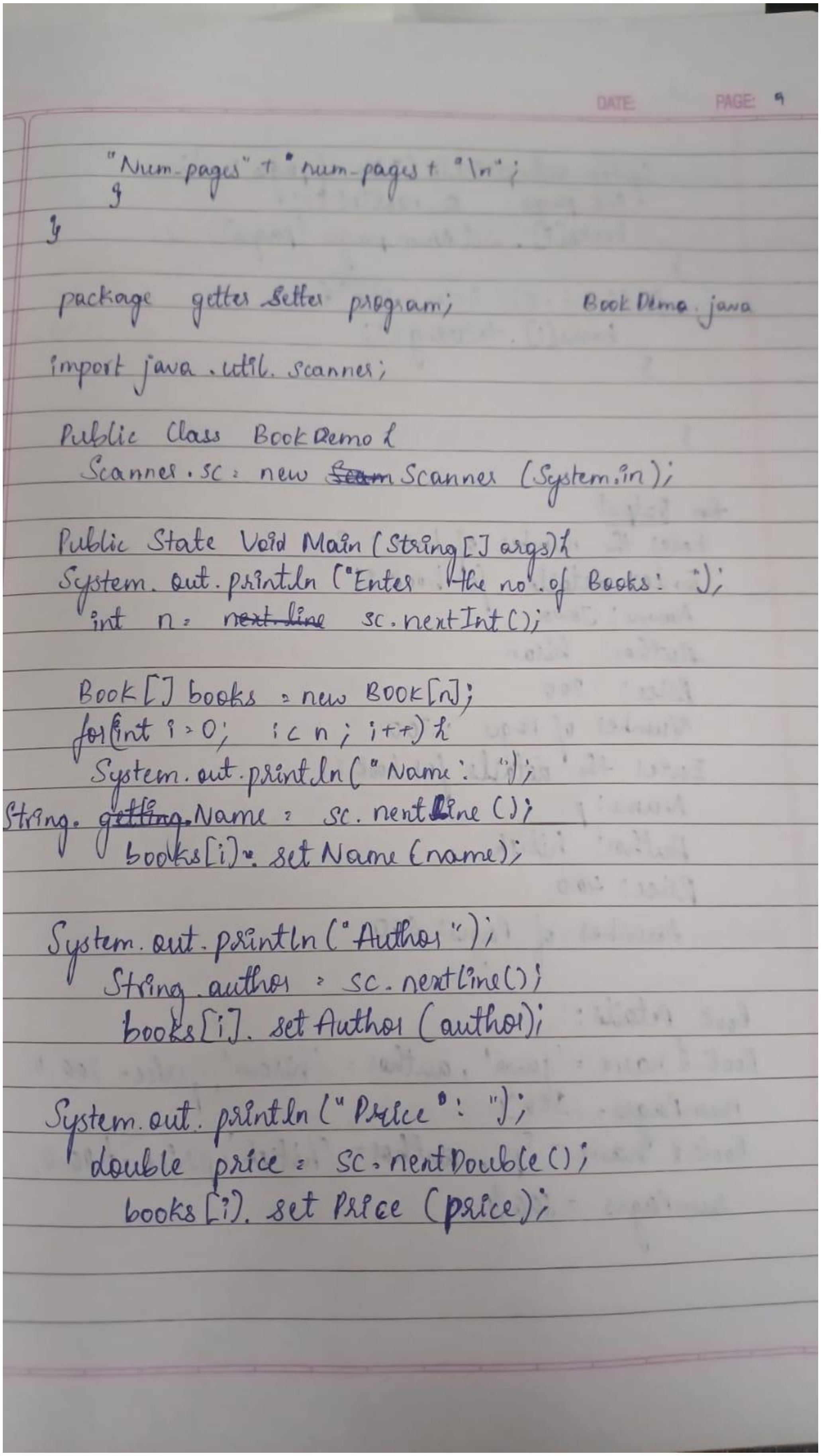
Create a class Book which contains four numbers : name, author, price , num-pages . Include a constructor to set the values for the ~~numbers~~ members. Include methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book . Develop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

~~Public Class~~

```
Public Class Book{
private String name;
private String author;
private double price;
private int num_pages;

 Public Book(String name, String author,
        double price, int num-pages){
        this.name = name.
        this. author= author
        this.price = price
        this.num_pages = num-pages
 }

  Public void SetName (String name){
     this.name = name;
   }

 Public String getName (){
      return Name;
 }
```

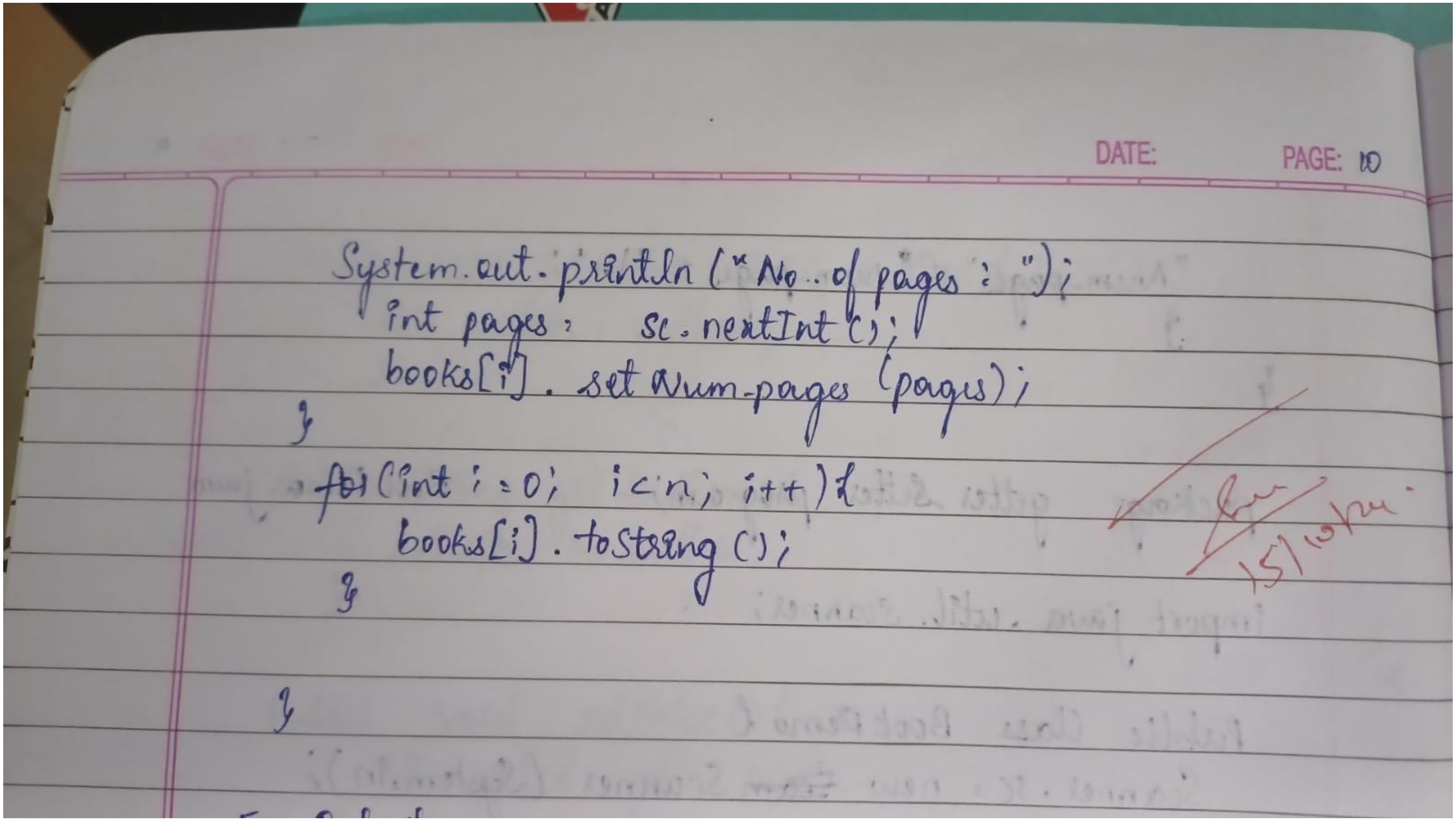


```
Public void SetAuthor (String Author){
   this.author = author;
  }

Public String get Author(){
     return ~~ow~~ author;
  }

Public void setPrice(){
     this.price = price;
  }

Public double getPrice(){
     return ~~RSD~~ price;
  }

Public void set Numpages (int num_pages){
  this.num_pages = num_pages.
 }

Public int get_Numpages (){
     return num_pages;
  }

@ Override

Public String toString(){
  return "Book Details" + ~~"\n"+~~ "\n" +
  ~~return~~ " Name" + name + "~~(\t)~~" "\n" +
  ~~return~~ " Author" + author + "\n" +
  ~~return~~ " Price" + price + "\n" +
```



DATE: PAGE: 9

```
     "Num-pages" + " num-pages + "\n";
     }
}

package getter Setter program;              Book Demo. java

import java. util. scanner;

Public Class BookDemo {
  Scanner . sc = new ~~Scan~~ Scanner (System.in);

Public State Void Main (String[] args){
System. out. println ("Enter the no. of Books: ");
   int n = ~~next line~~ sc. nextInt();

   Book[] books = new BOOK[n];
   for(int i = 0;   i < n ; i++) {
    System.out.println("Name: ");
String. ~~getting.~~Name = sc. nextLine ();
      books[i]. set Name (name);

System. out. println ("Author ");
   String. author = sc. nextLine();
     books[i]. set Author (author);

System.out. println ("Price": ");
  double price = sc. nextDouble();
     books[i]. set Price (price);
```



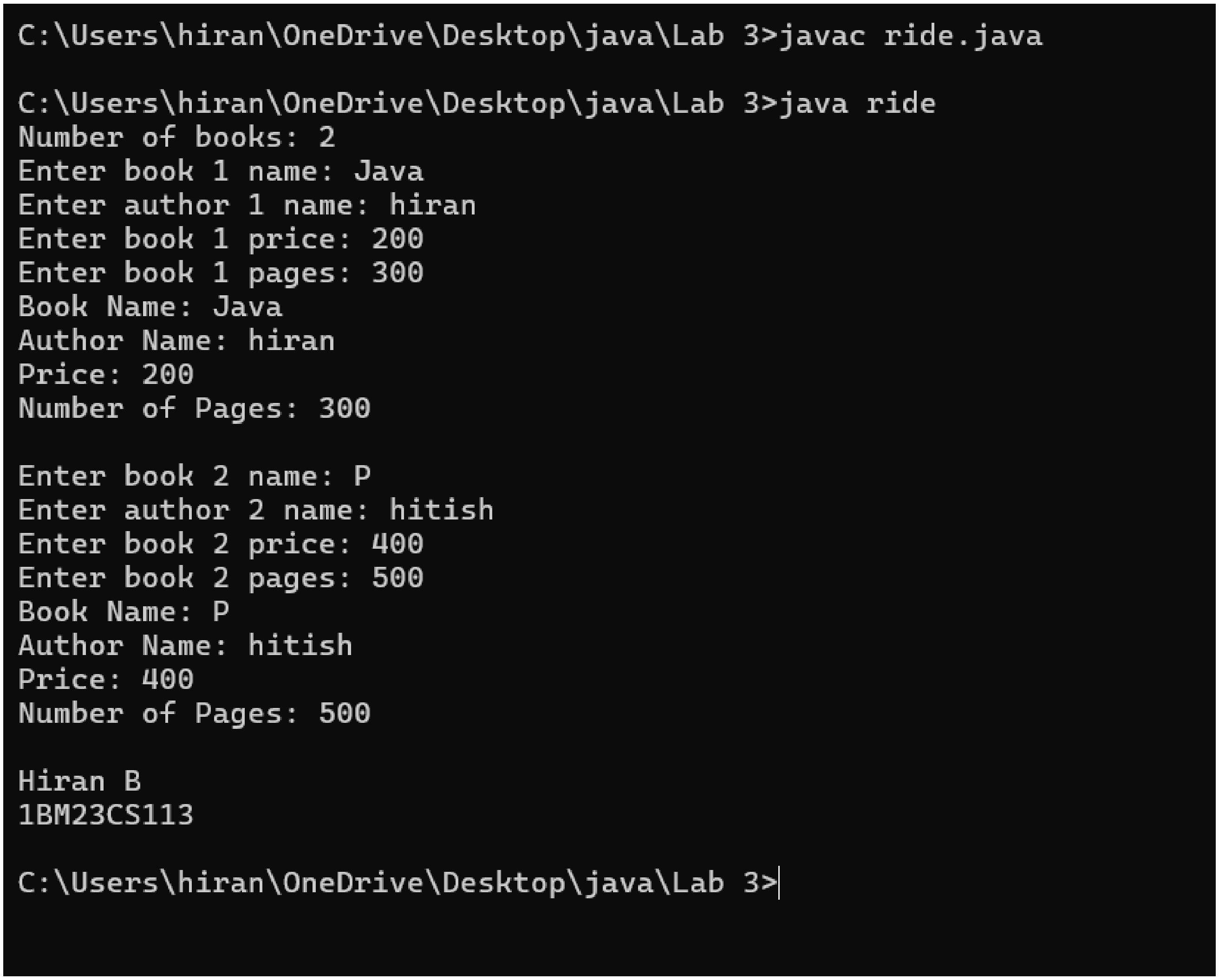
DATE: PAGE: 10

```
System.out.println("No..of pages : ");
    int pages = sc.nextInt();
    books[i].setNum-pages(pages);
}
for(int i=0; i<n; i++){
    books[i].toString();
  }

}
```

15/10/24

Code:

```
import java.util.Scanner;

class Book {
    public String book_name;
    public String author_name;
    public int price;
    public int num_pages;

    Book(String book_name, String author_name, int price, int num_pages) {
        this.book_name = book_name;
        this.author_name = author_name;
        this.price = price;
        this.num_pages = num_pages;
    }

    @Override
    public String toString() {
        String name, author, price, numPages;
        name = "Book Name: " + this.book_name + "\n";
        author = "Author Name: " + this.author_name + "\n";
        price = "Price: " + this.price + "\n";
        numPages = "Number of Pages: " + this.num_pages + "\n";
        return name + author + price + numPages;
    }
}

public class ride {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Number of books: ");
        int count = sc.nextInt();
        sc.nextLine();

        Book[] arr = new Book[count];
        for (int i = 0; i < count; i++) {
            System.out.print("Enter book " + (i + 1) + " name: ");
            String name = sc.nextLine();
            System.out.print("Enter author " + (i + 1) + " name: ");
            String author = sc.nextLine();
            System.out.print("Enter book " + (i + 1) + " price: ");
            int price = sc.nextInt();
            System.out.print("Enter book " + (i + 1) + " pages: ");
            int pages = sc.nextInt();
            sc.nextLine();

            arr[i] = new Book(name, author, price, pages);
            System.out.println(arr[i]);
        }

        sc.close();

        System.out.println("Hiran B ");
        System.out.println("1BM23CS113");
    }
}
```

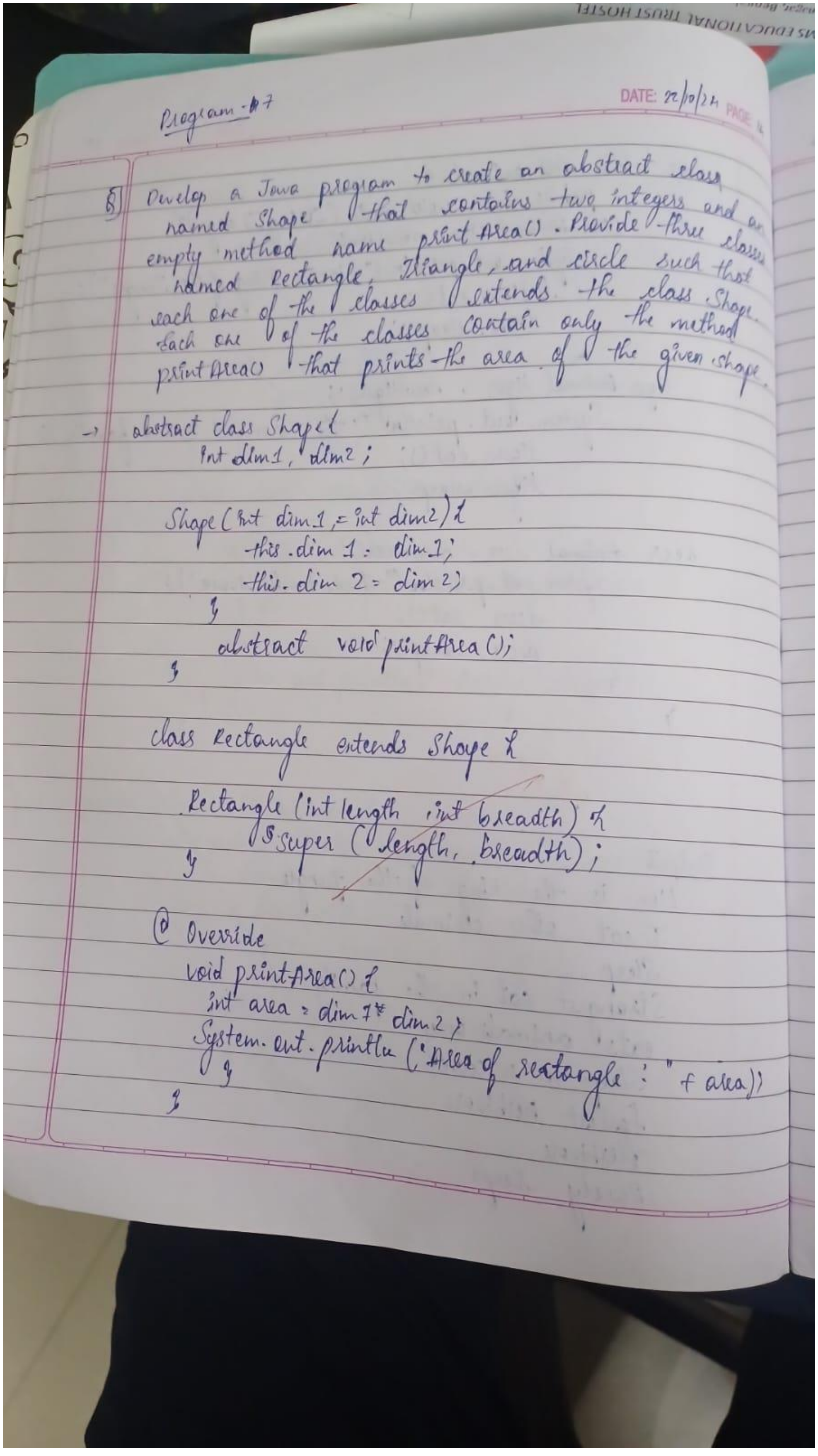


```
C:\Users\hiran\OneDrive\Desktop\java\Lab 3>javac ride.java

C:\Users\hiran\OneDrive\Desktop\java\Lab 3>java ride
Number of books: 2
Enter book 1 name: Java
Enter author 1 name: hiran
Enter book 1 price: 200
Enter book 1 pages: 300
Book Name: Java
Author Name: hiran
Price: 200
Number of Pages: 300

Enter book 2 name: P
Enter author 2 name: hitish
Enter book 2 price: 400
Enter book 2 pages: 500
Book Name: P
Author Name: hitish
Price: 400
Number of Pages: 500

Hiran B
1BM23CS113

C:\Users\hiran\OneDrive\Desktop\java\Lab 3>
```

Program 4

Abstract Classes

Algorithm:

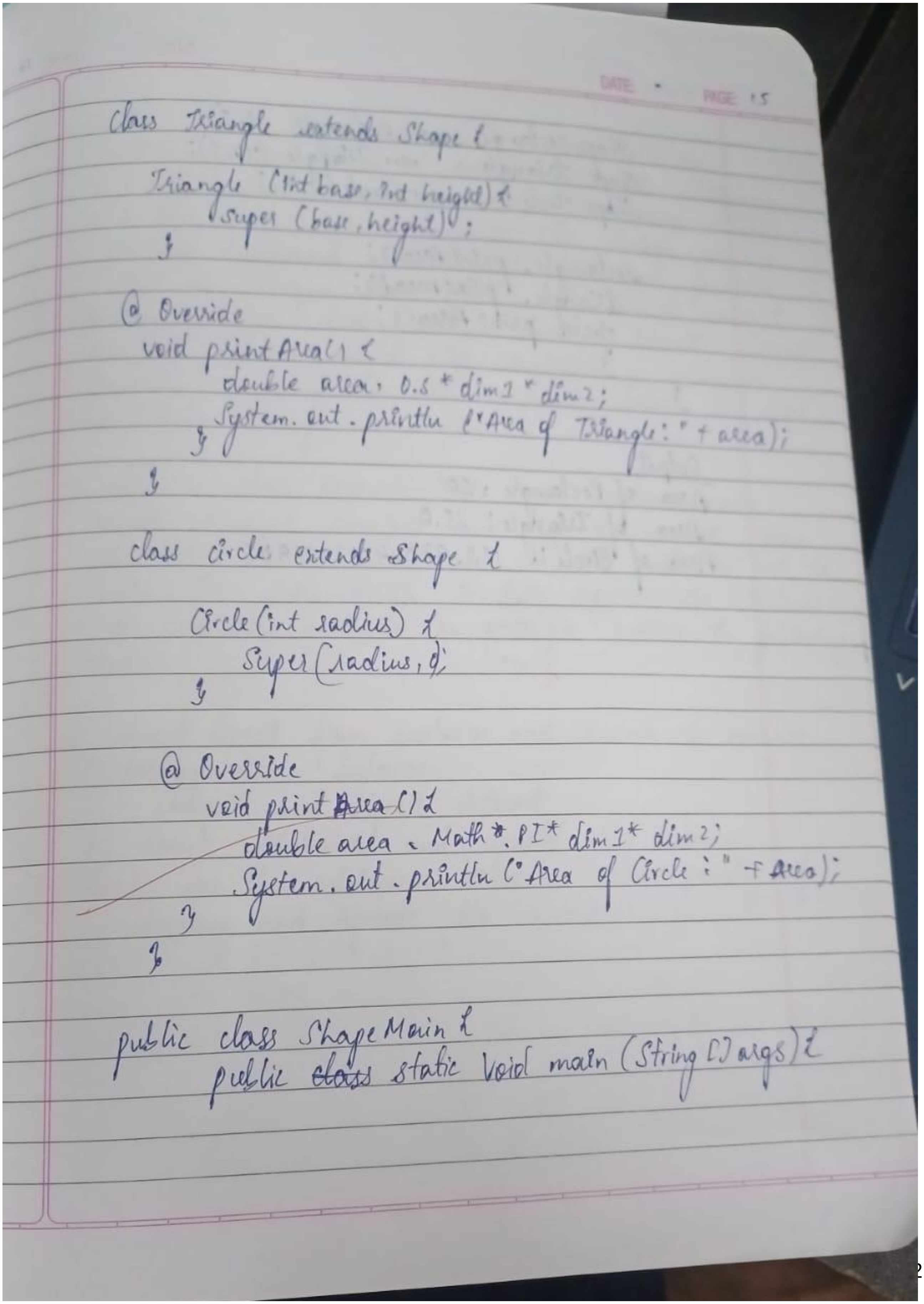


Program - 7

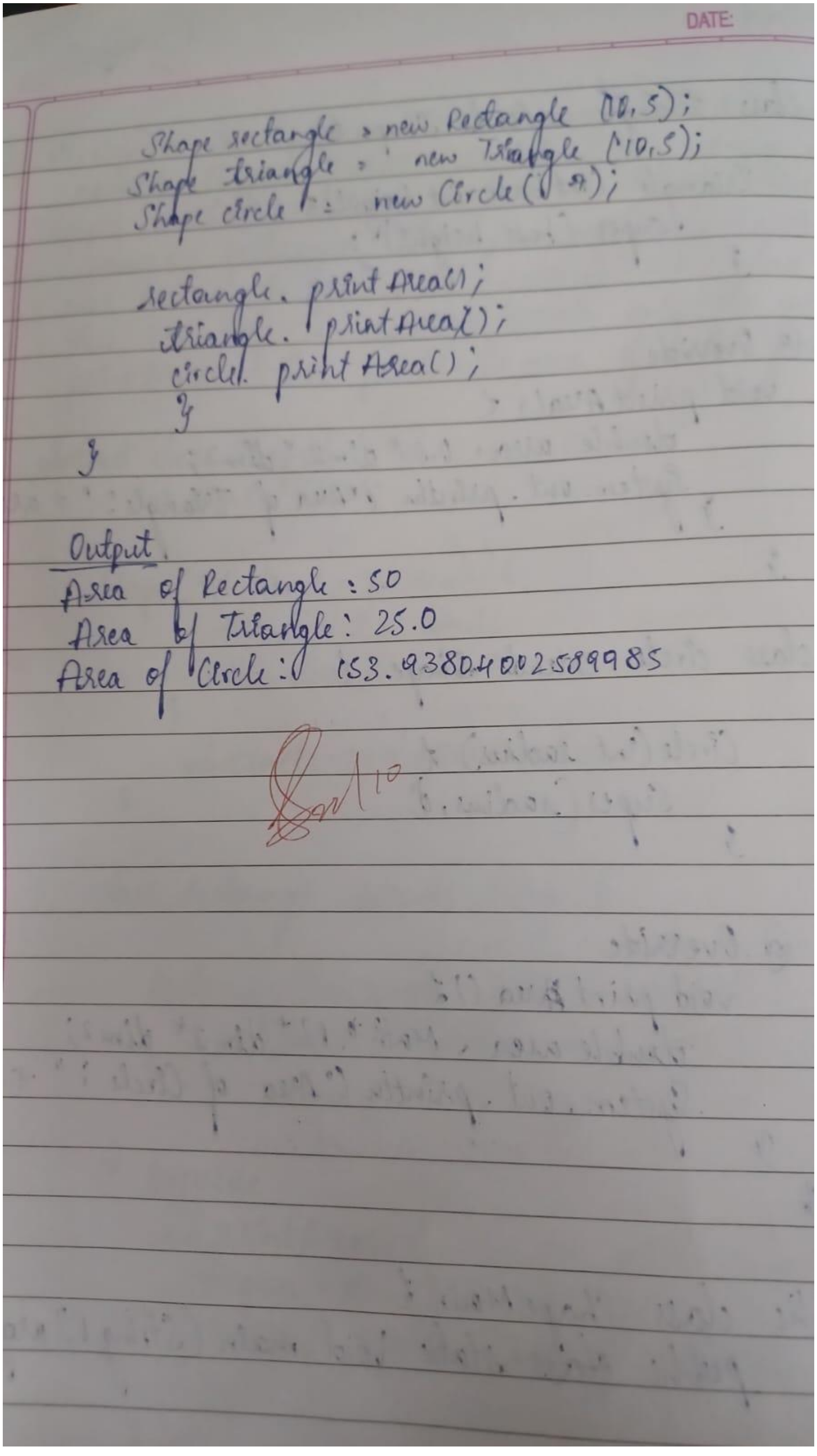
DATE: 22/10/24

8] Develp a Java program to create an abstract class named Shape that contains two integers and an empty method name print Area(). 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.

```
-> abstract class Shape{
        int dim1, dim2;

     Shape(int dim1,= int dim2){
          this.dim 1 = dim1;
          this.dim 2 = dim 2;
       }
       abstract void printArea();
    }

    class Rectangle extends Shape {

       Rectangle (int length ,int breadth) {
           Ssuper (length, breadth);
       }

    @ Override
       void printArea() {
        int area = dim 1* dim 2;
       System.out.println ("Area of reatangle : " + area);
        }
    }
```



```
class Triangle extends Shape {
   Triangle (int base, int height) {
       super (base, height);
   }

@ Override
 void printArea() {
        double area = 0.5 * dim1 * dim2;
        System.out.println ("Area of Triangle: " + area);
     }
 }

class circle extends Shape {

      Circle (int radius) {
          Super (radius, 0);
      }

   @ Override
      void print Area () {
          double area = Math ~~*~~. PI* dim1* dim2;
          System.out.println ("Area of Circle : " + Area);
      }
 }

public class ShapeMain {
       public ~~class~~ static Void main (String [] args) {
```



DATE:

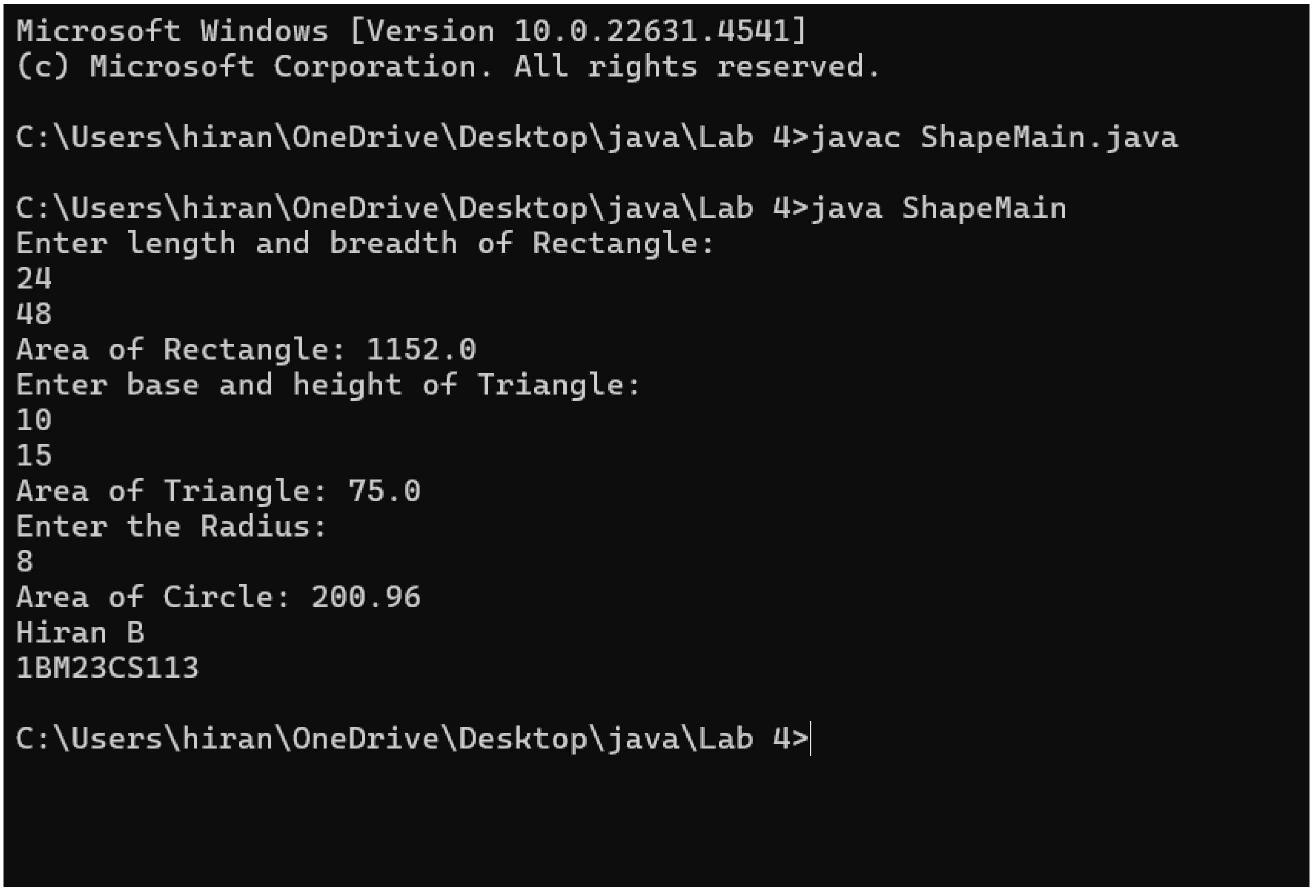
```
Shape rectangle = new Rectangle (10,5);
Shape triangle = new Triangle (10,5);
Shape circle = new Circle(7);

rectangle.printArea();
 triangle.printArea();
 circle.printArea();
 }
}
```

Output.

```
Area of Rectangle : 50
Area of Triangle : 25.0
Area of Circle : 153.93804002589985
```

Code:

```
import java.util.Scanner;

abstract class Shape {
    double dim1;
    double dim2;

    abstract void printarea();
}

class Rectangle extends Shape {
    Rectangle(double d1, double d2) {
        this.dim1 = d1;
        this.dim2 = d2;
    }

    @Override
    void printarea() {
        double area = dim1 * dim2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    Triangle(double base, double height) {
        this.dim1 = base;
        this.dim2 = height;
    }

    @Override
    void printarea() {
        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {
    Circle(double radius) {
        this.dim1 = radius;
    }

    @Override
    void printarea() {
        double area = 3.14 * dim1 * dim1;
        System.out.println("Area of Circle: " + area);
    }
}

public class area {
    public static void main(String[] args) {
        try (Scanner sc = new Scanner(System.in)) {
            System.out.println("Enter length and breadth of Rectangle:");
            double rl = sc.nextDouble();
            double rb = sc.nextDouble();
            Rectangle r1 = new Rectangle(rl, rb);
            r1.printarea();

            System.out.println("Enter base and height of Triangle:");
            double base = sc.nextDouble();
            double height = sc.nextDouble();
            Triangle t1 = new Triangle(base, height);
            t1.printarea();

            System.out.println("Enter the Radius:");
            double radius = sc.nextDouble();
            Circle c1 = new Circle(radius);
            c1.printarea();
        }

        System.out.println("Hiran B");
        System.out.println("1BM23CS113");
    }
}
```

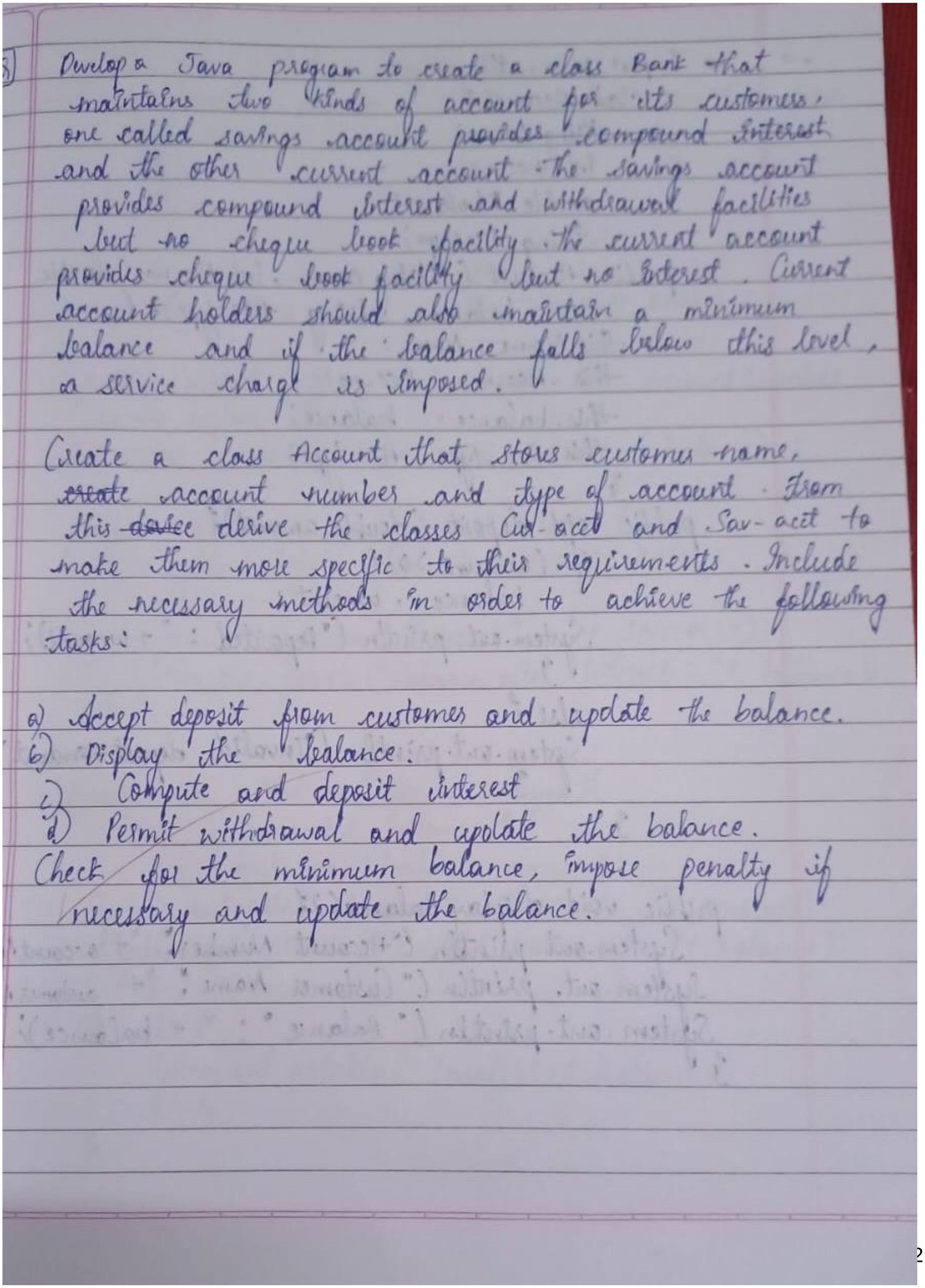


```
Microsoft Windows [Version 10.0.22631.4541]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hiran\OneDrive\Desktop\java\Lab 4>javac ShapeMain.java

C:\Users\hiran\OneDrive\Desktop\java\Lab 4>java ShapeMain
Enter length and breadth of Rectangle:
24
48
Area of Rectangle: 1152.0
Enter base and height of Triangle:
10
15
Area of Triangle: 75.0
Enter the Radius:
8
Area of Circle: 200.96
Hiran B
1BM23CS113

C:\Users\hiran\OneDrive\Desktop\java\Lab 4>
```

Program 5

Bank Account

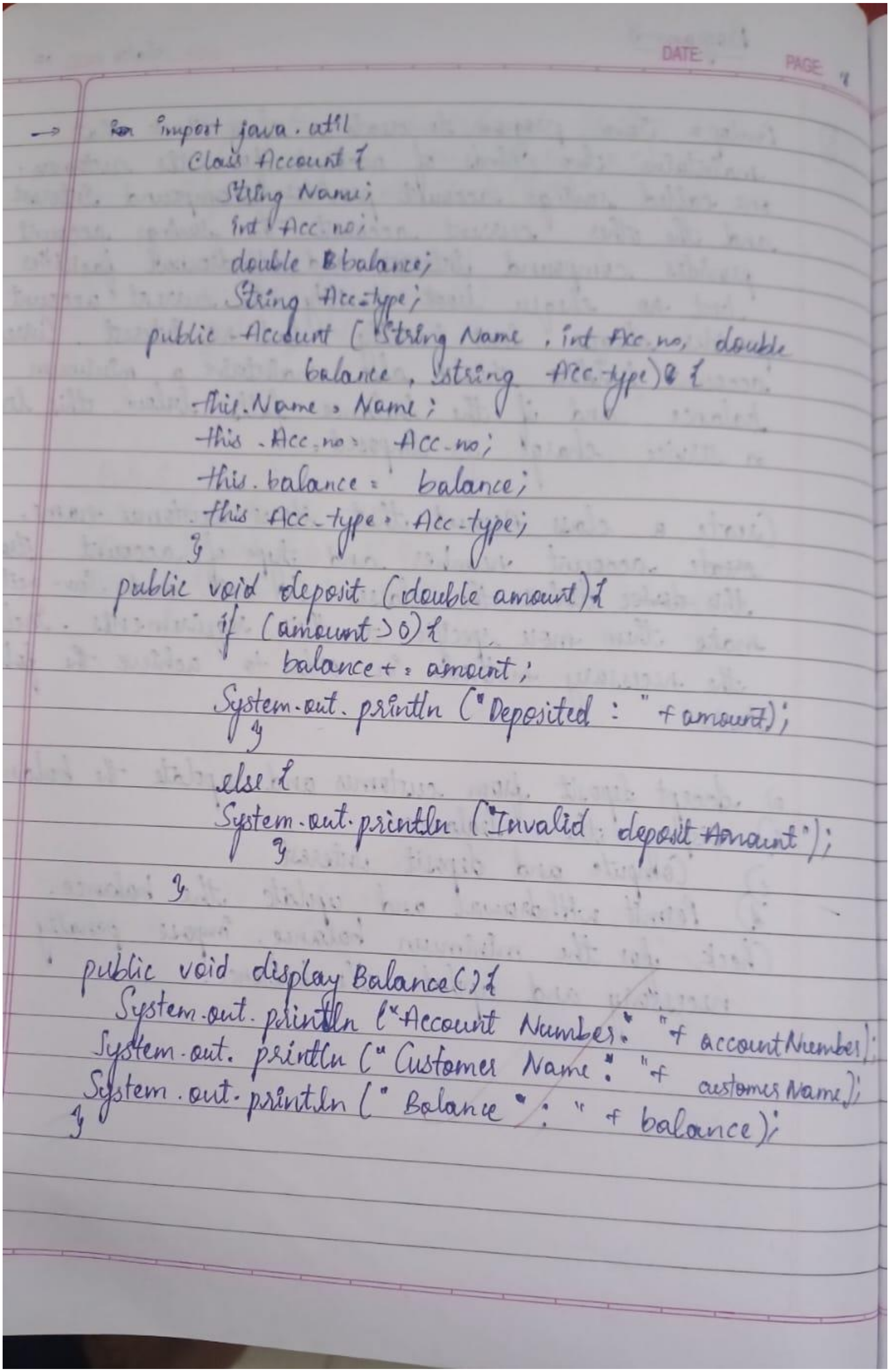
Algorithm:



5) Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account provides compound interest 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.

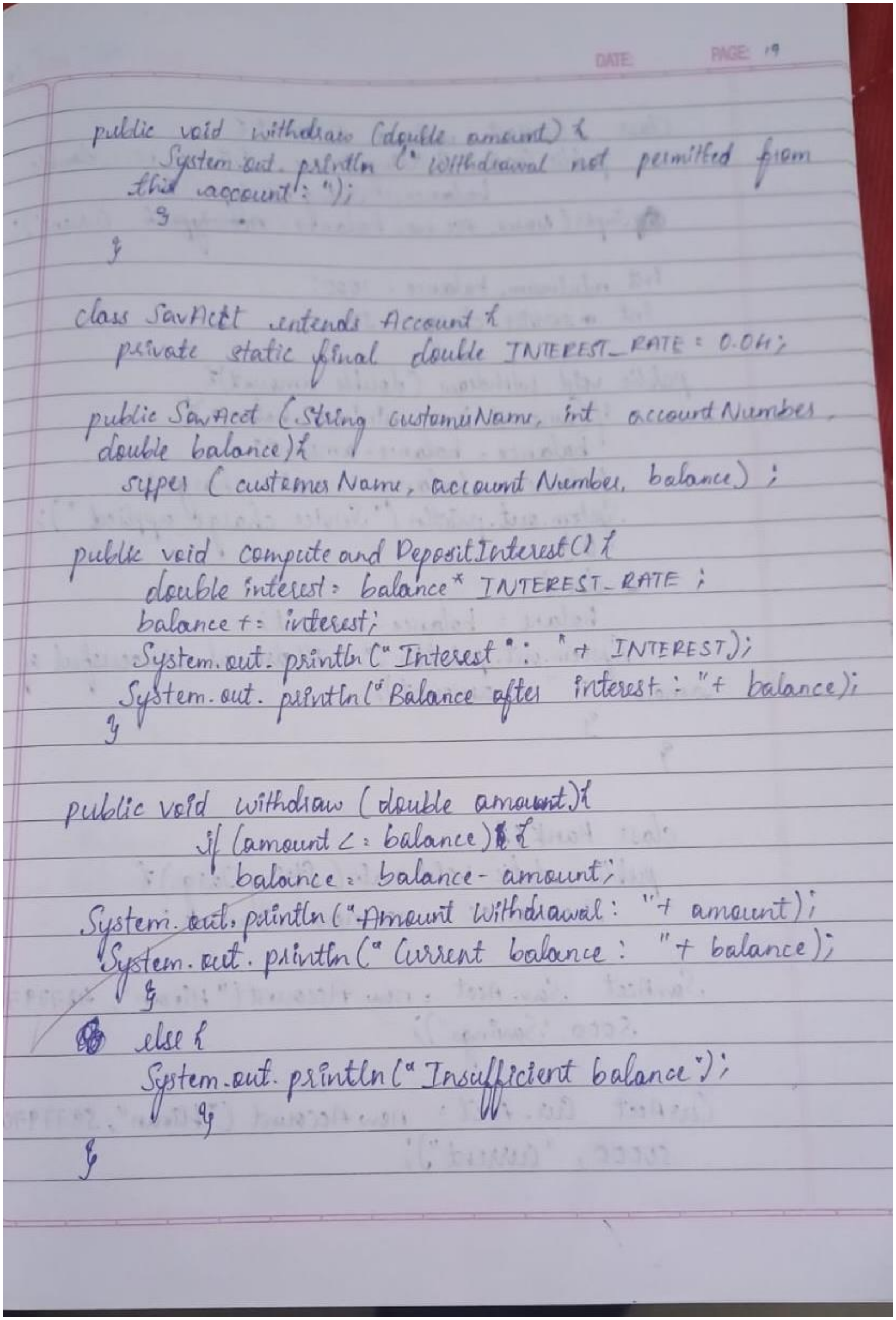
Create a class Account that stores customer name, ~~create~~ account number and type of account. From this ~~device~~ derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

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.



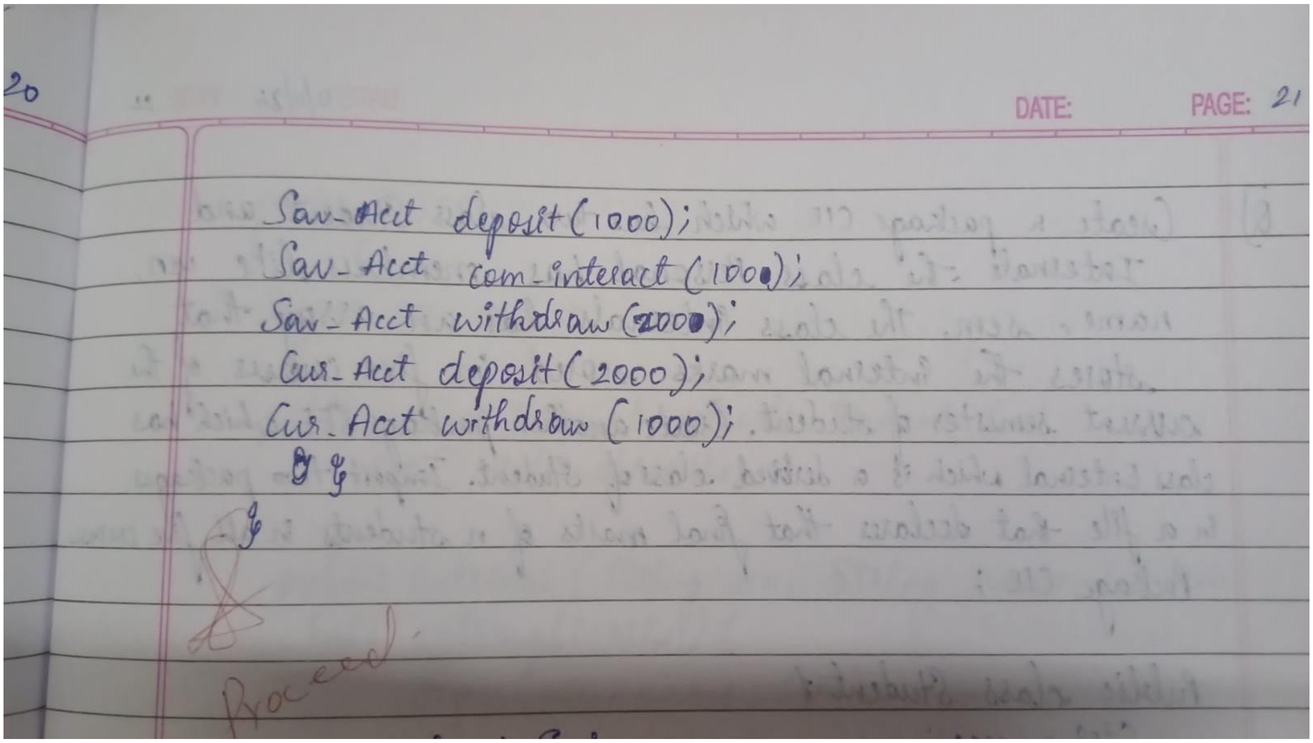
DATE: PAGE

```
→   ~~for~~ import java.util
        Class Account {
            String Name;
            int Acc.no;
            double ~~B~~balance;
            String Acc-type;
        public Account (String Name , int Acc.no, double
                    balance , String Acc-type) {
            this.Name = Name;
            this.Acc.no = Acc-no;
            this.balance = balance;
            this Acc-type = Acc-type;
        }
    public void deposit (double amount){
            if (amount > 0){
                balance += amount;
            System.out.println ("Deposited : " + amount);
            }
            else {
            System.out.println ("Invalid deposit Amount");
            }
        }

  public void display Balance(){
     System.out.println ("Account Number: " + accountNumber);
   System.out. println (" Customer Name: " + customer Name);
   System.out.println (" Balance : " + balance);
  }
```



```
public void withdraw (double amount) {
    System.out.println ("Withdrawal not permitted from
 this account: ");
    }
}

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

public SavAcct (String customerName, int accountNumber,
double balance){
   super ( customerName, accountNumber, balance) ;

public void computeandDepositInterest() {
      double interest = balance* INTEREST_RATE ;
      balance += interest;
      System.out.println (" Interest : "+ INTEREST);
     System.out.println(" Balance after interest : "+ balance);
 }

public void withdraw (double amount){
          if (amount <= balance) {
             balance = balance- amount;
 System.out.println (" Amount Withdrawal: "+ amount);
 System.out.println (" Current balance : "+ balance);
     }
  else {
     System.out.println (" Insufficient balance");
       }
}
```

```
Class CurAcct extend Account {
        public CurAcct ( String Name, int Acc.no. , double
                    balance, String Acc.type ) {
    Super( Name, Acc.no, balance, ~~Acc.type~~ "Current");

    int minimum-balance = 1000;
    Int service-charge = 50;

public void withdraw (double amount){
        if(amount > = minimum-balance){
         balance = balance - amount;
         balance = balance - service_charge;
        System.out.println ("Service charge applied ");
        }
    else {
         balane = balance - amount;
        System.out.println (" Withdrawal succesful of
amount :: " + amount);
         }
    }

    class Bank {
     public static void main( String[]args) {
         Scanner sc = new Scanner;

   SavAcct Sav-Acct = new Account(" Hiran", 4877970,
      5000, "Savings");

   CurAcct Cur-Acct = new Account ("Hiran", 5977970,
    50000, "Current");
```



20

DATE: PAGE: 21

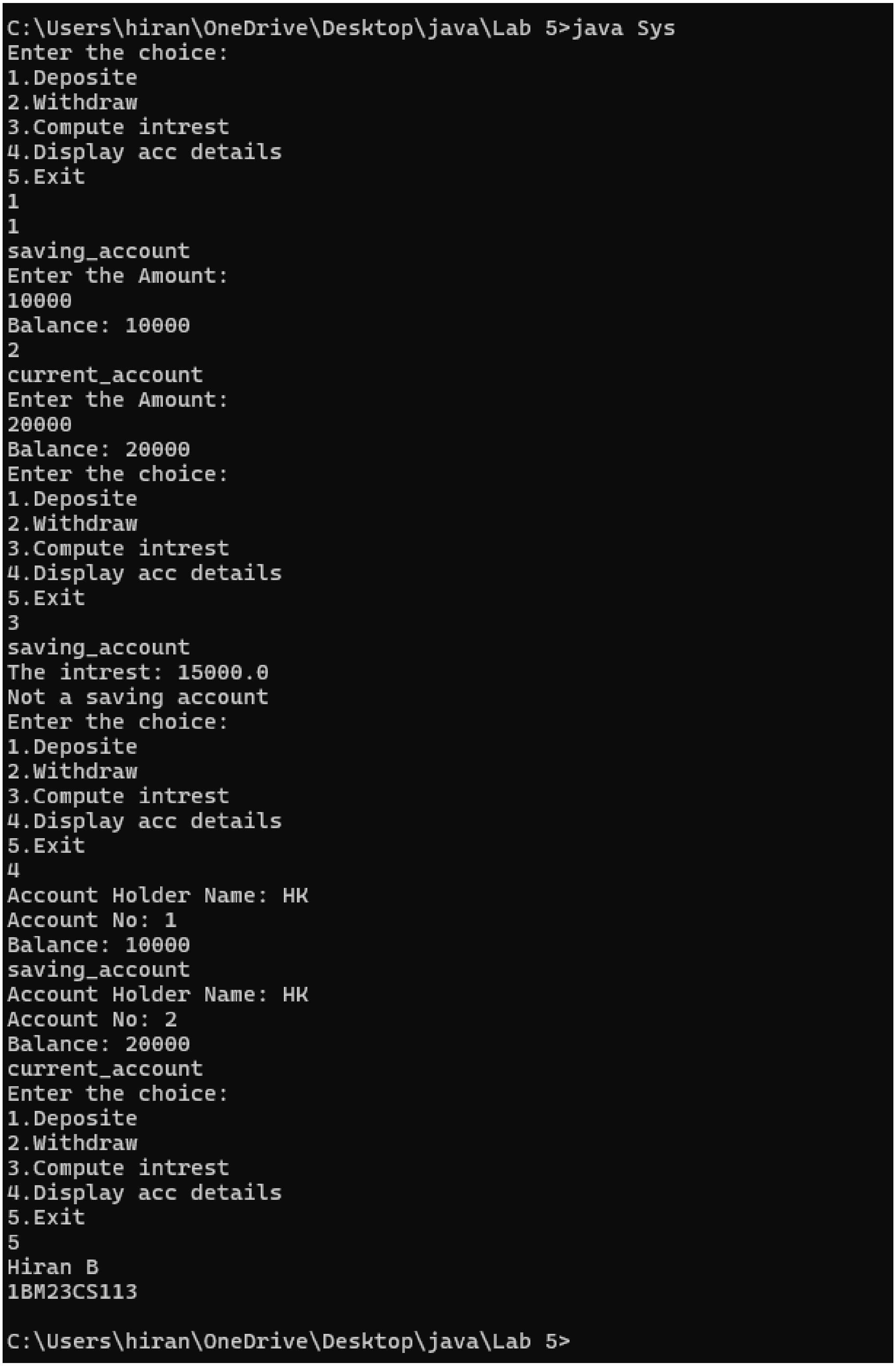
```
Sav_Acct deposit(1000);
Sav_Acct com_interact(1000);
Sav_Acct withdraw(2000);
Cur_Acct deposit(2000);
Cur_Acct withdraw(1000);
 } }
}
```

Proceed.

Code:

```
import java.util.Scanner;

class Account{
  Scanner sc=new Scanner(System.in);

  String name="HK";
  int money;
  String type;
  int accno;
  Account(String acctype,int accno){
    this.type=acctype;
    this.money=0;
    this.accno=accno;

  }
  void accdetail(){
    System.out.println("Account Holder Name: "+name);
    System.out.println("Account No: "+accno);
  System.out.println("Balance: "+money);
  System.out.println(this.type);
}

void deposit(){
  int mon;
  System.out.println(accno);
  System.out.println(type);
  System.out.println("Enter the Amount: ");
  mon=sc.nextInt();
  money+=mon;
  System.out.println("Balance: "+money);
}

void withdraw(){
  System.out.println(this.accno);
  System.out.println(type);
  int mon;
  System.out.println("Enter the Amount: ");
  mon=sc.nextInt();
  money-=mon;
  System.out.println("Balance: "+money);
  if((money<=100) && this.type=="current_account")
  {
    System.out.println("Minimum balance is 100");
    System.out.println("Deposite money now and pay the fine of 50");
  }
}
void cal_intrest(){
  if(this.type=="saving_account")
  {

    System.out.println(this.type);
    double temp=this.money;

    double intrest=((temp)*0.5)+temp;
    System.out.println("The intrest: "+intrest);
  }

  else
```

```
    {
      System.out.println("Not a saving account");
    }

  }
}

public class Sys {
    public static void main(String[] args) {
      Account c1=new Account("saving_account",1);
      Account c2=new Account("current_account",2);

      while(true)
      {
        Scanner sc=new Scanner(System.in);

        int choice;
        System.out.println("Enter the choice:\n1.Deposite\n2.Withdraw\n3.Compute intrest\n4.Display acc details\n5.Exit");
        choice=sc.nextInt();
        if (choice==1)
        {
          c1.deposit();
          c2.deposit();
        }

        if(choice==2){
           c1.withdraw();
           c2.withdraw();
        }

        if(choice==3){
          c1.cal_intrest();
          c2.cal_intrest();
       }

       if(choice==4){
        c1.accdetail();
        c2.accdetail();
     }
     if(choice==5){
      break;
      }
    }

    System.out.println("Hiran B ");
    System.out.println("1BM23CS113");
  }
}
```



```
C:\Users\hiran\OneDrive\Desktop\java\Lab 5>java Sys
Enter the choice:
1.Deposite
2.Withdraw
3.Compute intrest
4.Display acc details
5.Exit
1
1
saving_account
Enter the Amount:
10000
Balance: 10000
2
current_account
Enter the Amount:
20000
Balance: 20000
Enter the choice:
1.Deposite
2.Withdraw
3.Compute intrest
4.Display acc details
5.Exit
3
saving_account
The intrest: 15000.0
Not a saving account
Enter the choice:
1.Deposite
2.Withdraw
3.Compute intrest
4.Display acc details
5.Exit
4
Account Holder Name: HK
Account No: 1
Balance: 10000
saving_account
Account Holder Name: HK
Account No: 2
Balance: 20000
current_account
Enter the choice:
1.Deposite
2.Withdraw
3.Compute intrest
4.Display acc details
5.Exit
5
Hiran B
1BM23CS113

C:\Users\hiran\OneDrive\Desktop\java\Lab 5>
```

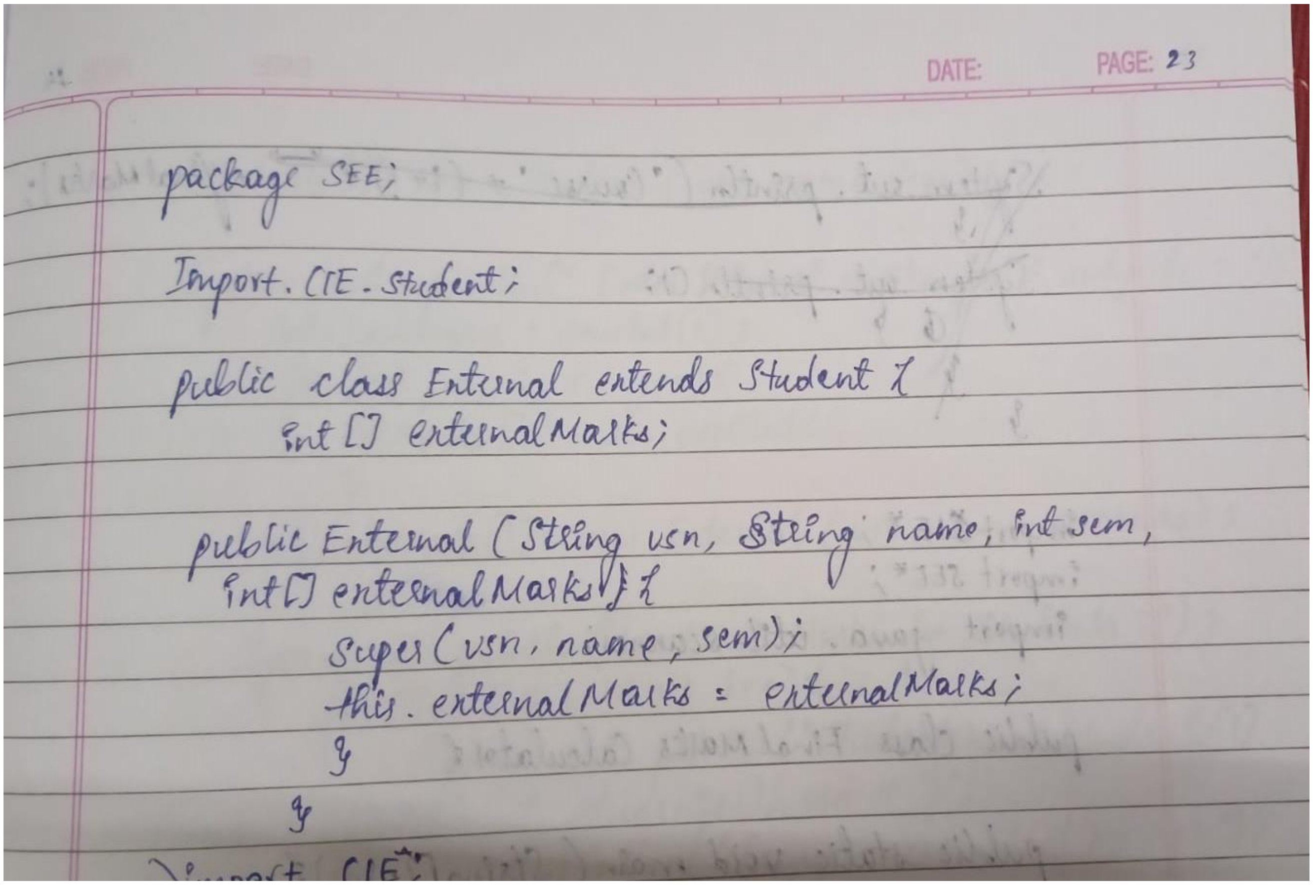
Program 6

Packages

Algorithm:

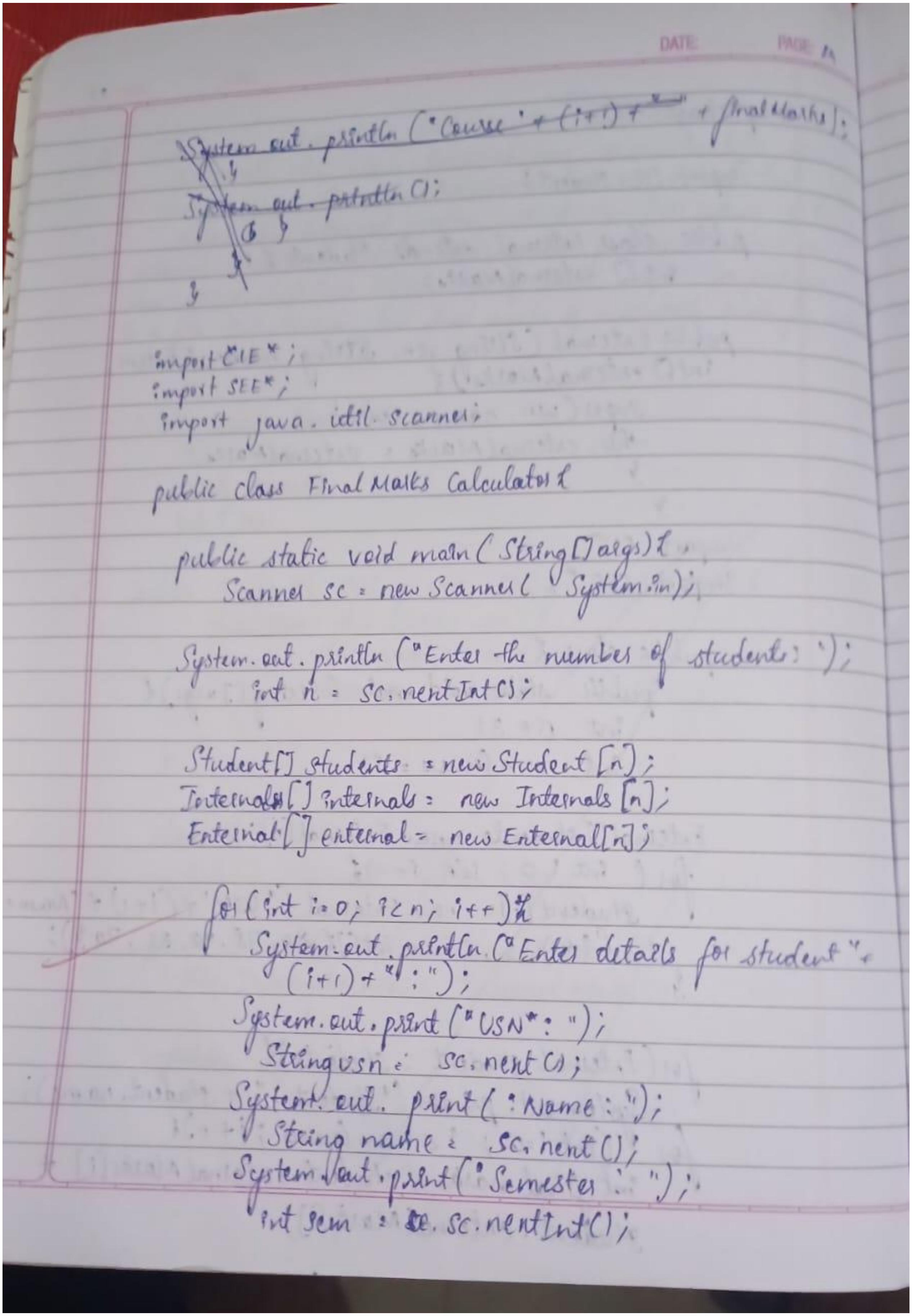
Q) Create a package CIE which has two class Student and Internals . the class . Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of student. Create another package SEE which has class External which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

```
Package CIE ;

Public class Student {
    String usN;
    String Name;
    int sem;

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

   class Internals {
      int [] internalMarks;

   public Internals (int[] InternalMarks) {
     this. InternalMarks : InternalMarks;
           }
   ~~public~~ }
```



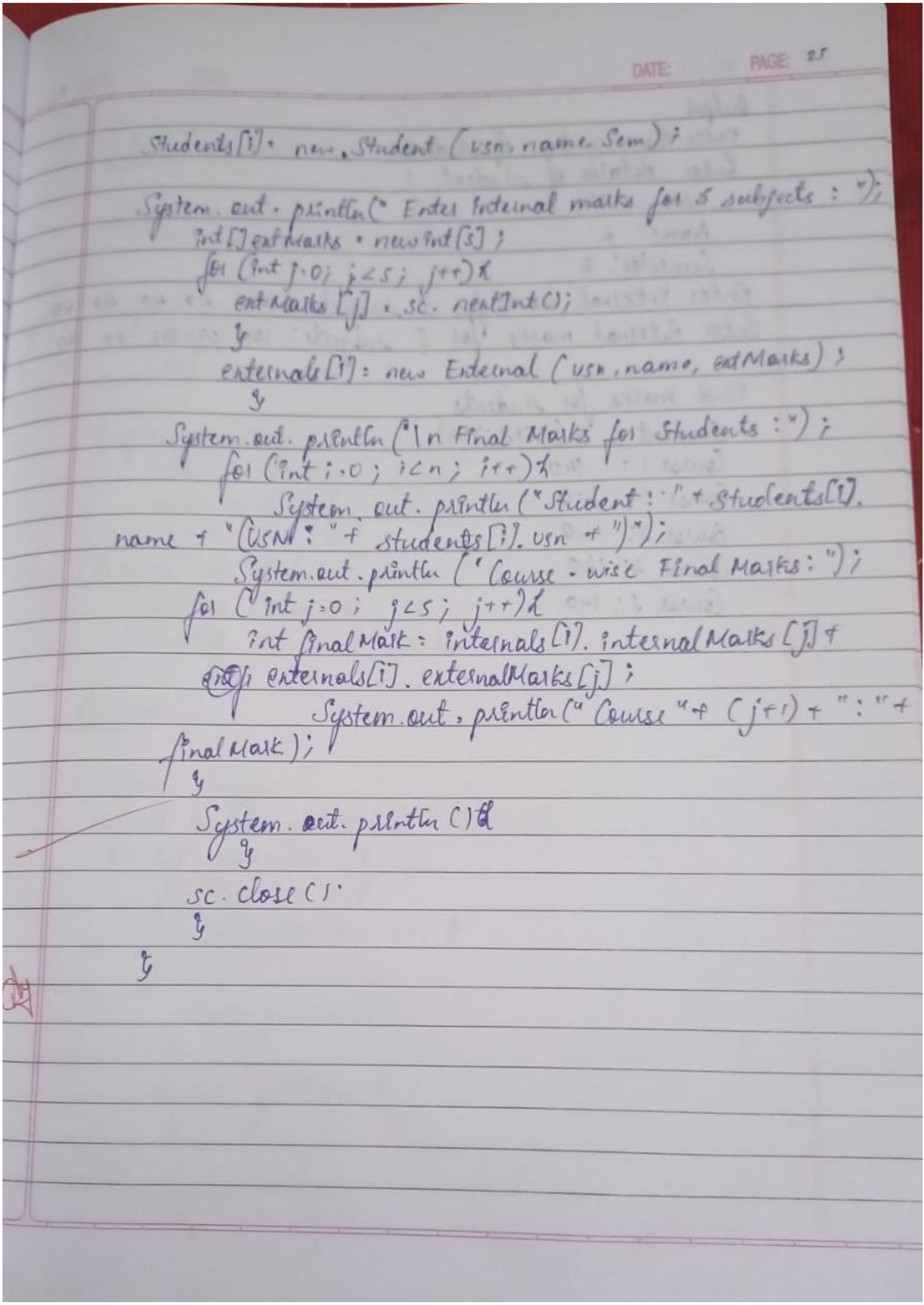
DATE: PAGE: 23

```
package SEE;

Import. CIE. Student;

public class External extends Student {
        int [] externalMarks;

  public External (String usn, String name, int sem,
   int[] externalMarks) {
          super(usn, name, sem);
          this. externalMarks = externalMarks;
          }
       }
import CIE
```



DATE PAGE 11

```
System out. println ("Course " + (i+1) + " " + finalMarks);
   }
System out. println ();
   } }
   }
}

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

public class Final Marks Calculator {

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

  System. out. println ("Enter the number of students: ");
      int n = sc. nextInt();

  Student[] students = new Student [n];
  Internals[] internals = new Internals [n];
  External[] external = new External[n];

   for (int i=0; i<n; i++){
      System.out. println ("Enter details for student "+
         (i+1) + " : ");
      System.out. print ("USN : ");
        String usn = sc. next ();
      System. out. print ( "Name : ");
        String name = sc. next ();
      System. out. print ("Semester : ");
       int sem = sc. nextInt();
```

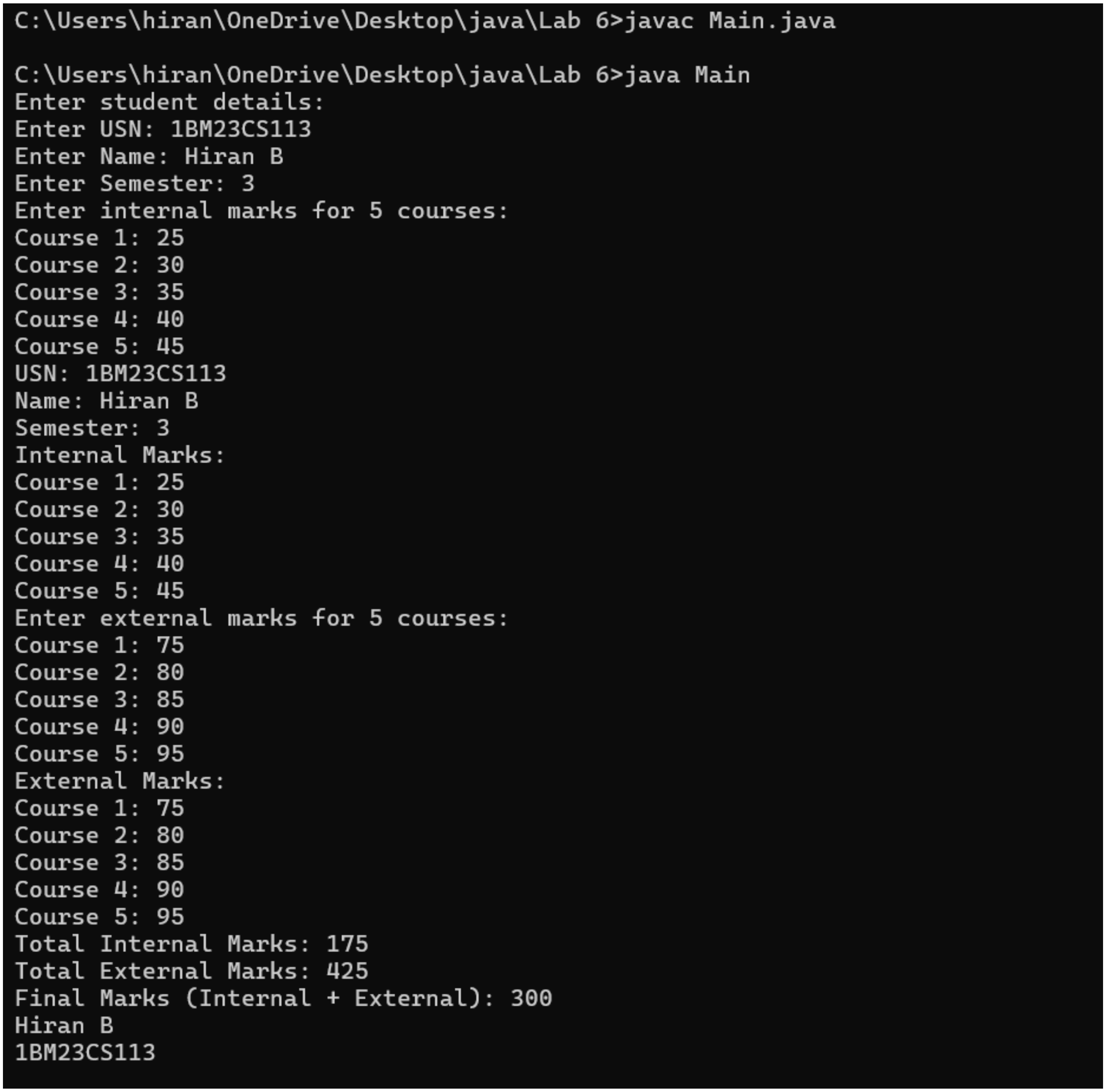


```
Students[i] = new Student (usn, name, Sem);

System.out.println(" Enter Internal marks for 5 subjects : ");
   int[] extMarks = new int[5];
   for (int j=0; j<5; j++){
      extMarks [j] = sc. nextInt();
      }
      externals[i] = new External (usn, name, extMarks);
        }
  System.out.println("\n Final Marks for Students :");
      for (int i=0; i<n; i++){
            System.out.println("Student : " + Students[i].
name + "(USN: " + students[i].usn + ")");
         System.out.println ("Course - wise Final Marks: ");
     for (int j=0; j<5; j++){
         int finalMark = internals[i]. internalMarks [j] +
     ~~int~~ externals[i]. externalMarks[j];
              System.out.println(" Course " + (j+1) + ": " +
finalMark);
      }
      System.out.println()
      }
      sc.close()
      }
  }
```

Code:

```
package CIE;

import java.util.Scanner;

public class Internals extends Student {

    int[] cieMarks = new int[5];

    public void inputCIEMarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter CIE marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            cieMarks[i] = s.nextInt();
        }
    }

    public int[] getCieMarks() {
        return cieMarks;
    }
}


package CIE;

import java.util.Scanner;

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

    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = s.nextLine();
        System.out.print("Enter Name: ");
        name = s.nextLine();
        System.out.print("Enter Semester: ");
        sem = s.nextInt();
    }

    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}

package SEE;

import CIE.Student;
import java.util.Scanner;

public class External extends Student {
    int[] seeMarks = new int[5];

    public void inputSEEMarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            seeMarks[i] = s.nextInt();
        }
    }

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


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

```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of students: ");
        int numStudents = sc.nextInt();
        sc.nextLine();

        Internals[] cieStudents = new Internals[numStudents];
        External[] seeStudents = new External[numStudents];

        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nEnter details for student " + (i + 1) + ":");

            cieStudents[i] = new Internals();
            cieStudents[i].inputStudentDetails();
            cieStudents[i].inputCIEMarks();

            seeStudents[i] = new External();
            seeStudents[i].inputSEEMarks();
        }

        System.out.println("\nFinal marks for each student:");
        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nDetails for student " + (i + 1) + ":");
            cieStudents[i].displayStudentDetails();

            int[] cieMarks = cieStudents[i].getCieMarks();
            int[] seeMarks = seeStudents[i].getSeeMarks();
            int[] finalMarks = new int[5];

            System.out.println("Final marks in each subject:");
            for (int j = 0; j < 5; j++) {
                finalMarks[j] = cieMarks[j] + seeMarks[j];
                System.out.println("Subject " + (j + 1) + ": " + finalMarks[j]);
            }
        }

        sc.close();
        System.out.println("Hiran B ");
        System.out.println("1BM23CS113");
    }
}
```

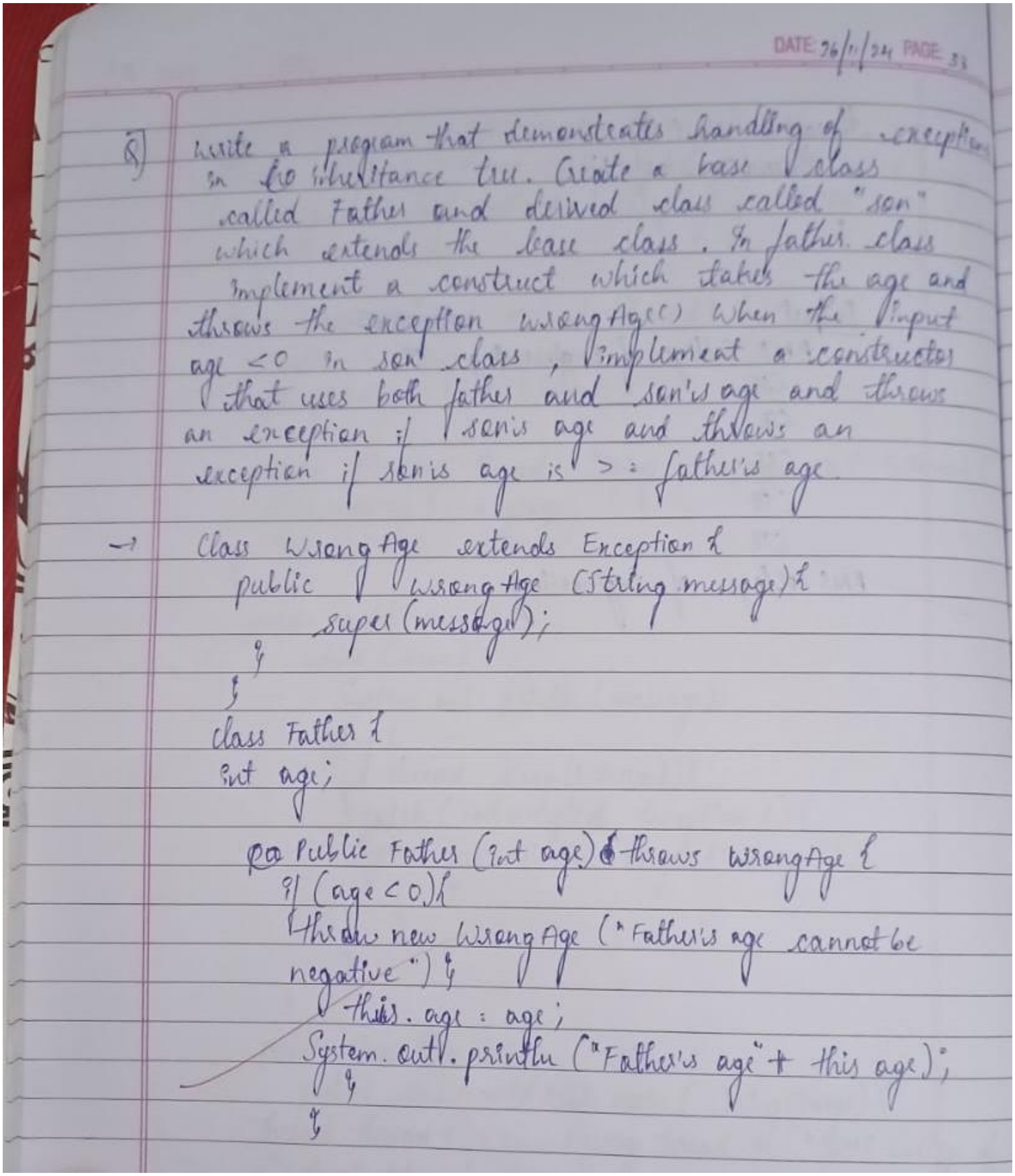


```
C:\Users\hiran\OneDrive\Desktop\java\Lab 6>javac Main.java

C:\Users\hiran\OneDrive\Desktop\java\Lab 6>java Main
Enter student details:
Enter USN: 1BM23CS113
Enter Name: Hiran B
Enter Semester: 3
Enter internal marks for 5 courses:
Course 1: 25
Course 2: 30
Course 3: 35
Course 4: 40
Course 5: 45
USN: 1BM23CS113
Name: Hiran B
Semester: 3
Internal Marks:
Course 1: 25
Course 2: 30
Course 3: 35
Course 4: 40
Course 5: 45
Enter external marks for 5 courses:
Course 1: 75
Course 2: 80
Course 3: 85
Course 4: 90
Course 5: 95
External Marks:
Course 1: 75
Course 2: 80
Course 3: 85
Course 4: 90
Course 5: 95
Total Internal Marks: 175
Total External Marks: 425
Final Marks (Internal + External): 300
Hiran B
1BM23CS113
```

<u>**Program 7**</u>

Exception handling

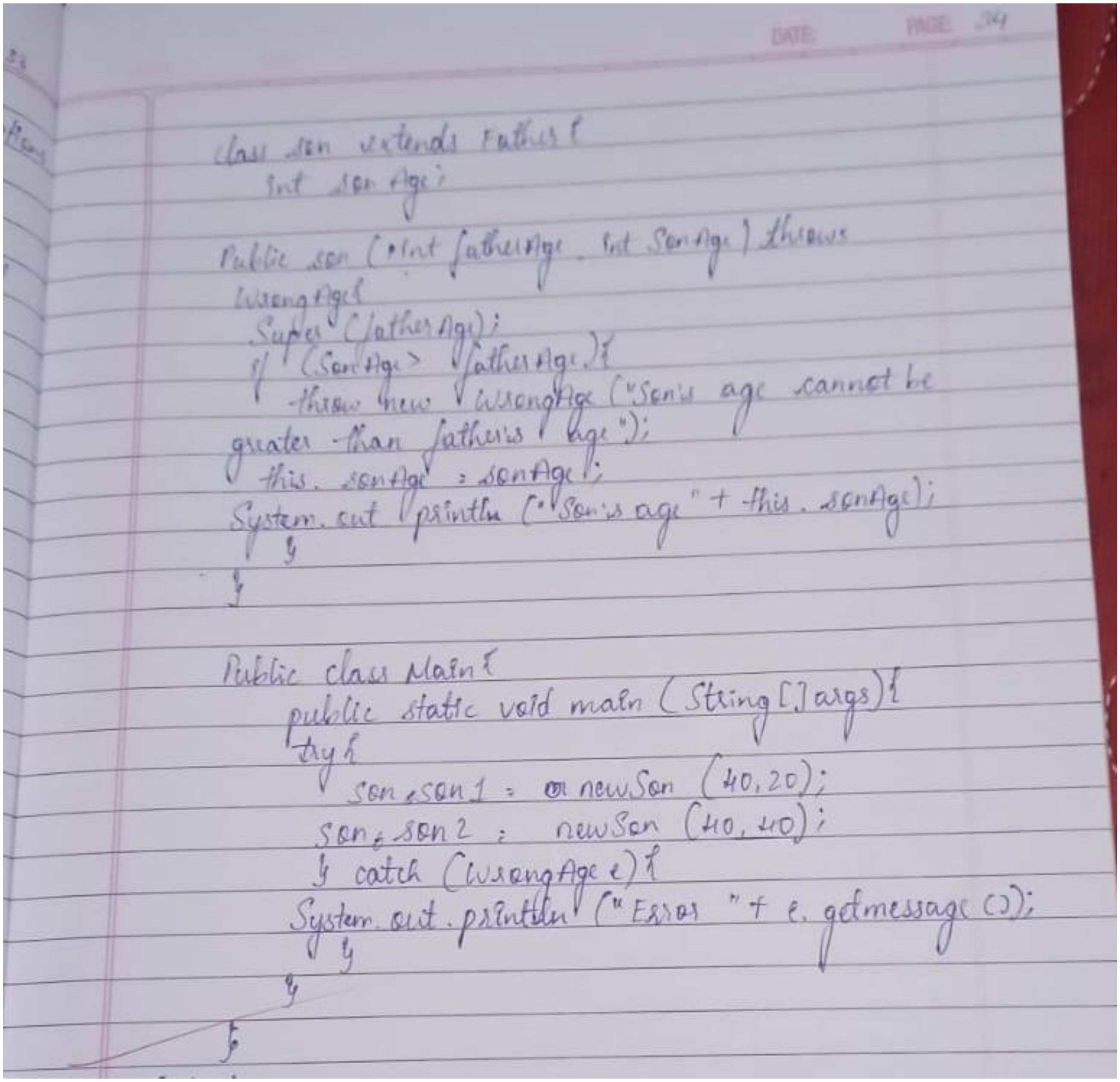
Algorithm:



DATE: 26/11/24 PAGE: 33

Q) write a program that demonstrates handling of exception in the inheritance tree. Create a base class called Father and derived class called "son" which extends the base class. In father class implement a construct 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 and throws an exception if son's age is >= father's age.

→
```
Class WrongAge extends Exception {
    public WrongAge (String message){
            super(message);
      }
  }
  class Father {
  int age;

      Public Father (int age) throws WrongAge {
        if (age < 0){
        throw new WrongAge ("Father's age cannot be
        negative") }
          this. age = age;
        System. out. println ("Father's age" + this age);
          }
        }
```



```
class son extends Father{
    int son Age;

Public son (int fatherAge, int SonAge) throws
 WrongAge{
  Super (fatherAge);
  if (SonAge > fatherAge){
    throw new WrongAge ("Son's age cannot be
greater than father's age");
   this. sonAge = sonAge;
  System.out println ("Son's age" + this. sonAge);
   }
 }

Public class Main{
    public static void main (String[]args){
    try{
       son son1 = newSon (40,20);
     son son2 = newSon (40, 40);
     } catch (WrongAge e){
    System.out.println ("Error "+ e. getmessage ());
      }
   }
}
```

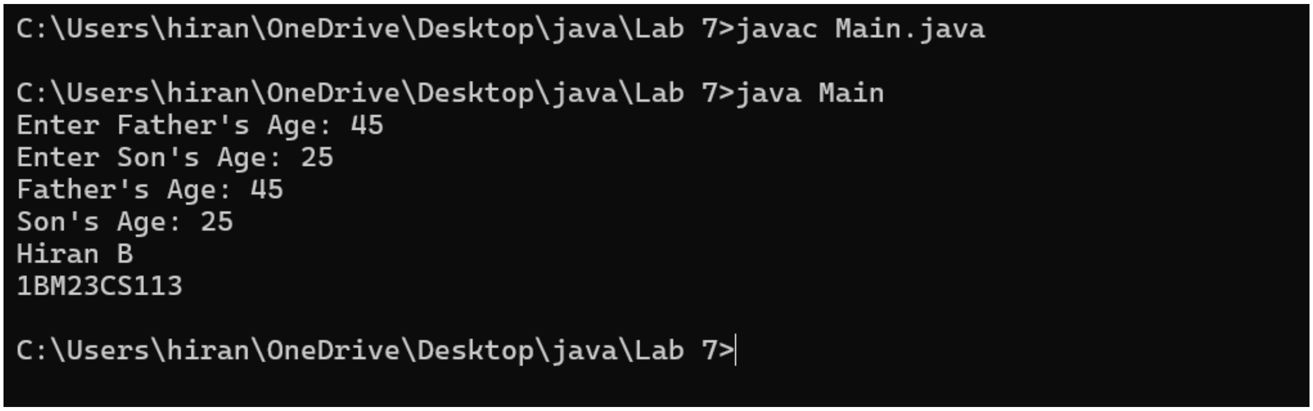
Code:

import java.util.Scanner;

class WrongAge extends Exception {

```
    public WrongAge() {
        super("Age Error");
    }

    public WrongAge(String message) {
        super(message);
    }
}

class Father {
    protected int fatherAge;

    public Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Father's Age: ");
        fatherAge = s.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

    public void display() {
        System.out.println("Father's Age: " + fatherAge);
    }
}

class Son extends Father {
    private int sonAge;

    public Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Son's Age: ");
        sonAge = s.nextInt();
        if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        } else if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to Father's
age");
```

```
        }
    }

    @Override
    public void display() {
        super.display();
        System.out.println("Son's Age: " + sonAge);
    }
}

public class Main {
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Exception Caught: " + e.getMessage());
        }

        System.out.println("Hiran B ");
        System.out.println("1BM23CS113");
    }
}
```



```
C:\Users\hiran\OneDrive\Desktop\java\Lab 7>javac Main.java

C:\Users\hiran\OneDrive\Desktop\java\Lab 7>java Main
Enter Father's Age: 45
Enter Son's Age: 25
Father's Age: 45
Son's Age: 25
Hiran B
1BM23CS113

C:\Users\hiran\OneDrive\Desktop\java\Lab 7>
```

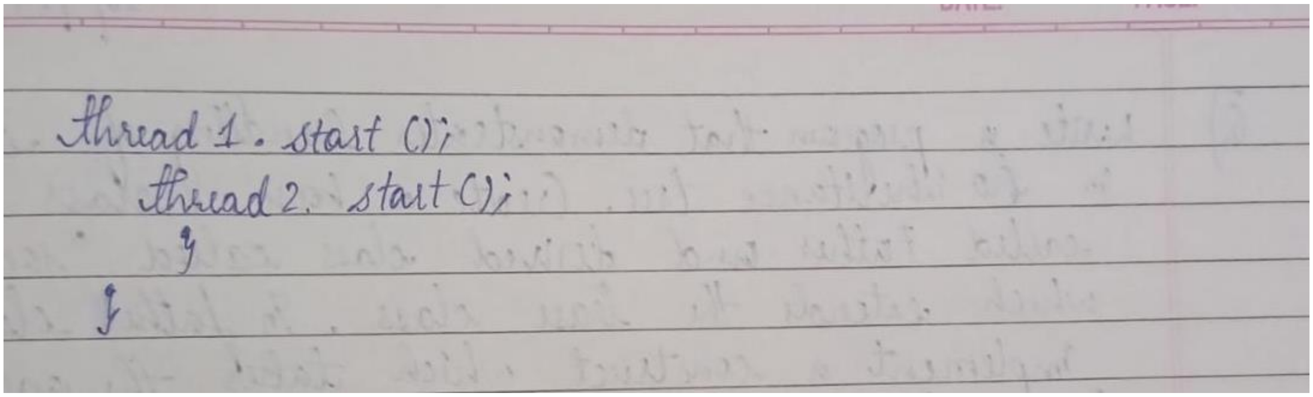
Program 8

Threads

Algorithm:

Q] write a program which creates two threads ~~as~~, one thread Displaying "BMS College of Engineering" once every ten seconds and attacks displaying "CSE" once every two seconds.

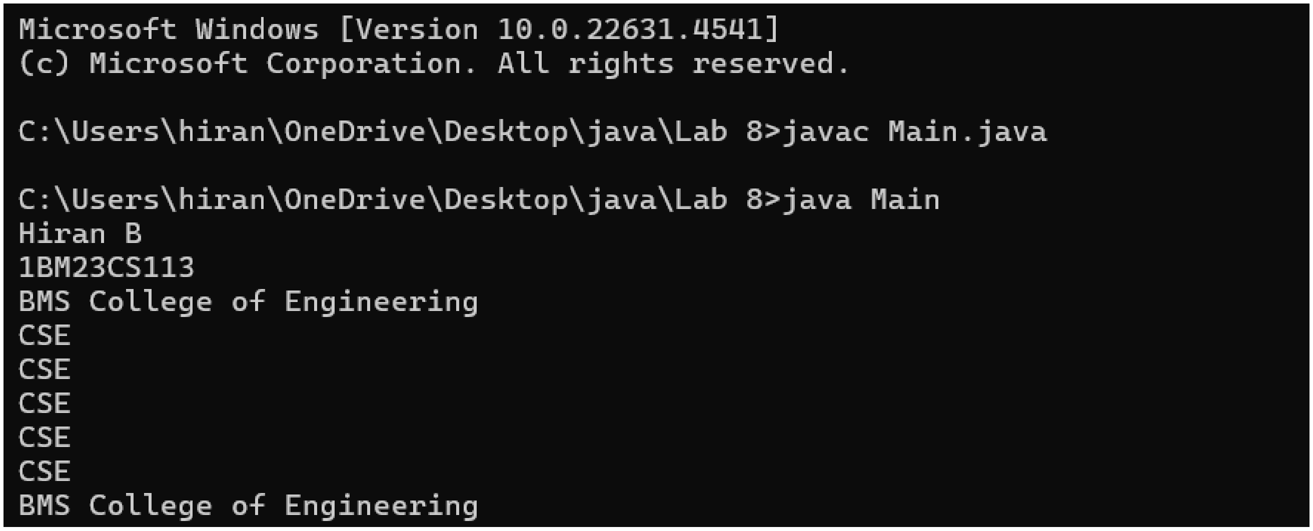
```
class MessageThread extends thread {
  private String message;
   ~~pri~~ private int interval;

  public MessageThread (String message, int interval){
   this.message = message;
   this.interval = interval;
  }
  public void run () {
     while (true) {
        System.out. println (message);
        try {
          Thread. sleep (interval);
        } catch (Interrupted Exception e) {
         System. out. println (e. get Message ());
            }
          }
       }
    }
   public class Main {
      public static void ~~Main~~ main ( String [] args) {
   Thread thread1 = new MessageThread ("BMS College of
   of Engineering", 10000);

   Thread thread 2 . new message Thread ("CSE", 2000);
```



```
Thread 1. start ();
  Thread 2. start ();
    }
}
```

Code:

```
class MessageThread extends Thread {
    private String message;
    private int interval;

    // Constructor for the thread
    public MessageThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }


    @Override
    public void run() {
        while (true) {
            System.out.println(message);
            try {
                Thread.sleep(interval);
            } catch (InterruptedException e) {
                System.err.println(e.getMessage());
            }
        }
    }
}

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

        Thread thread1 = new MessageThread("BMS College of Engineering", 10000); //
10 seconds
        Thread thread2 = new MessageThread("CSE", 2000); // 2 seconds


        thread1.start();
        thread2.start();
        System.out.println("Hiran B");
        System.out.println("1BM23CS113");
    }
}
```

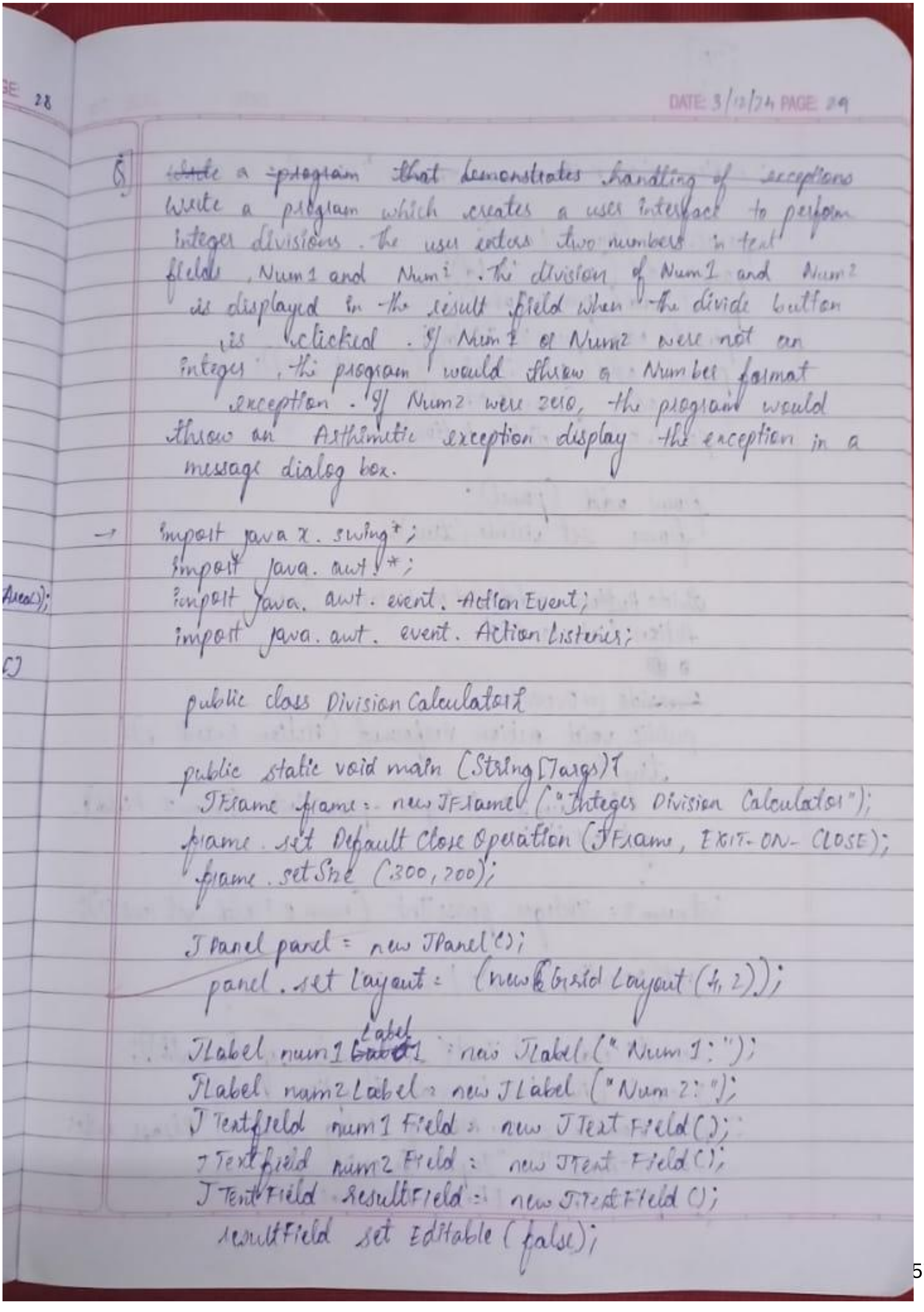


```
Microsoft Windows [Version 10.0.22631.4541]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hiran\OneDrive\Desktop\java\Lab 8>javac Main.java

C:\Users\hiran\OneDrive\Desktop\java\Lab 8>java Main
Hiran B
1BM23CS113
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
```

Program 9

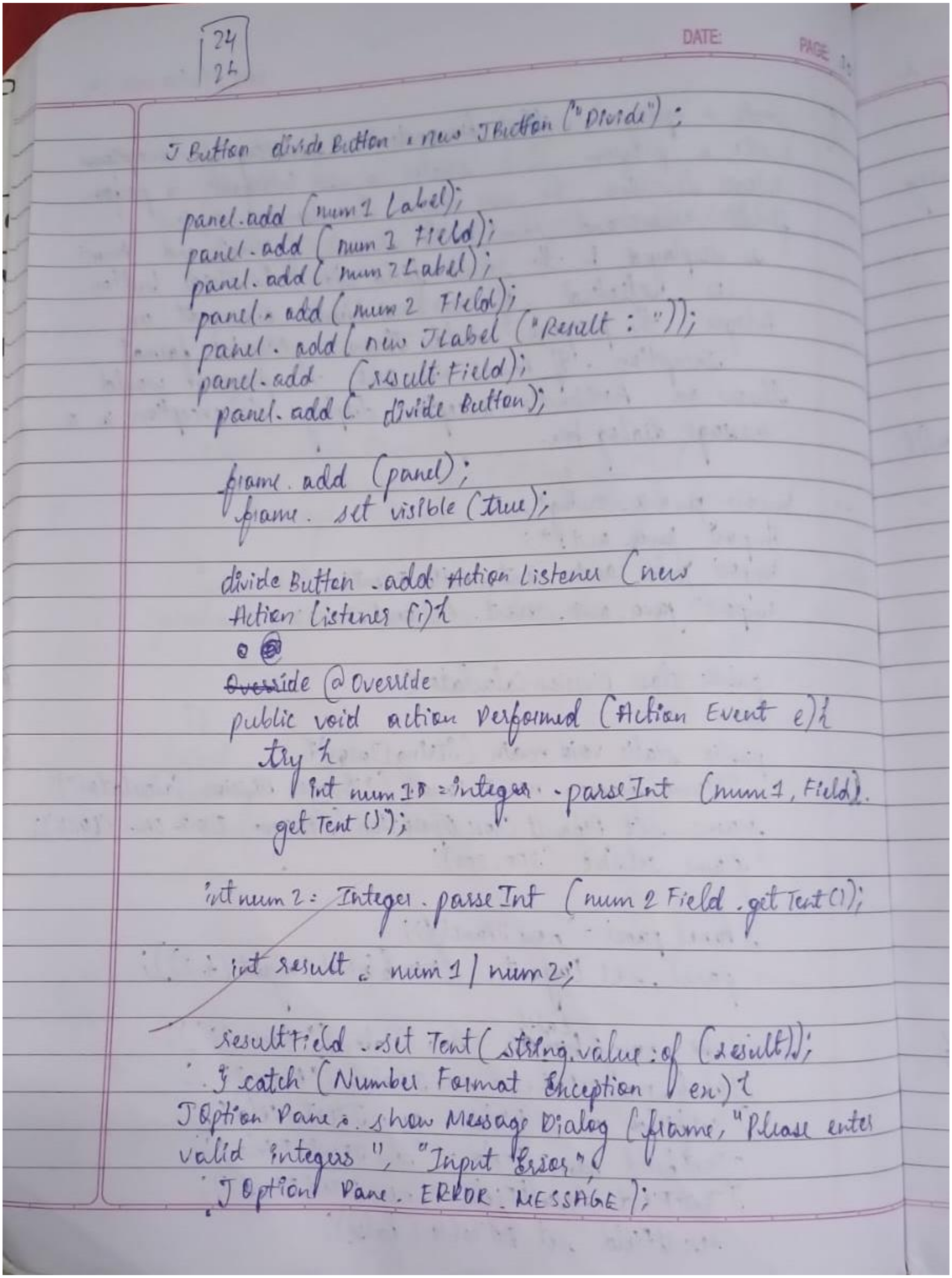
Calculator

Algorithm:



Q] ~~Write a program that demonstrates handling of exceptions~~ Write a program which creates a user interface to perform integer divisions. the user enters two numbers in 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 Number format exception. If Num2 were zero, the program would throw an Arthimetic exception display the exception in a message dialog box.

```
→ import javax.swing*;
  import java.awt.*;
  import java.awt.event.ActionEvent;
  import java.awt.event.ActionListener;

  public class DivisionCalculator{

  public static void main (String[]args){
   JFrame frame = new JFrame ("Integer Division Calculator");
  frame.setDefaultCloseOperation (JFrame, EXIT-ON-CLOSE);
   frame.setSize (300,200);

  JPanel panel = new JPanel();
   panel.setLayout = (new GridLayout (4,2));

  JLabel num1Label = new JLabel ("Num 1:");
  JLabel num2Label = new JLabel ("Num 2:");
  JTextfield num1Field = new JTextField();
  JTextfield num2Field = new JTextField();
  JTextField resultField = new JTextField();
   resultField setEditable (false);
```

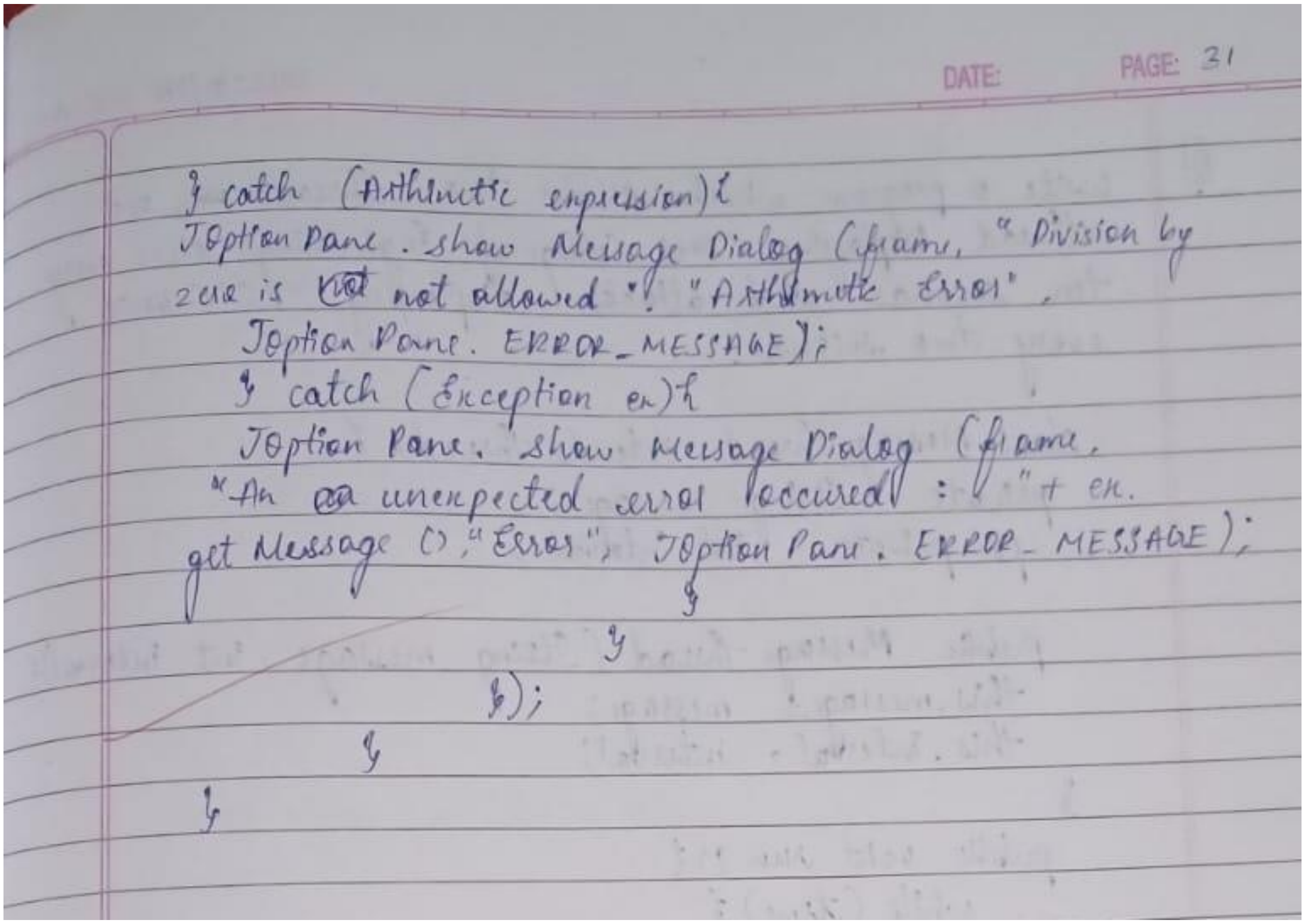


24
24

DATE:

PAGE

```
J Button divide Button = new JButton ("Divide");

panel.add (num1 Label);
panel.add ( num1 Field);
panel.add ( num2Label);
panel. add ( num 2 Field);
panel. add ( new JLabel ("Result : "));
panel.add (result Field);
 panel. add ( divide Button);

  frame. add (panel);
  frame. set visible (true);

  divide Button .add Action Listener (new
  Action Listener ()){
   @ @
  ~~Over~~ride @ Override
  public void action Performed (Action Event e){
    try {
     int num 1 = integer .parseInt (num1, Field).
   get Text ()");

int num 2 = Integer . parseInt (num 2 Field . get Text ());

 int result = num 1 / num 2;

  resultField . set Text( string.value:of (result));
  } catch (Number Format Exception ex){
JOption Pane . show Message Dialog (frame, "Please enter
valid integers ", "Input Error",
 JOptionl Pane. ERROR . MESSAGE );
```



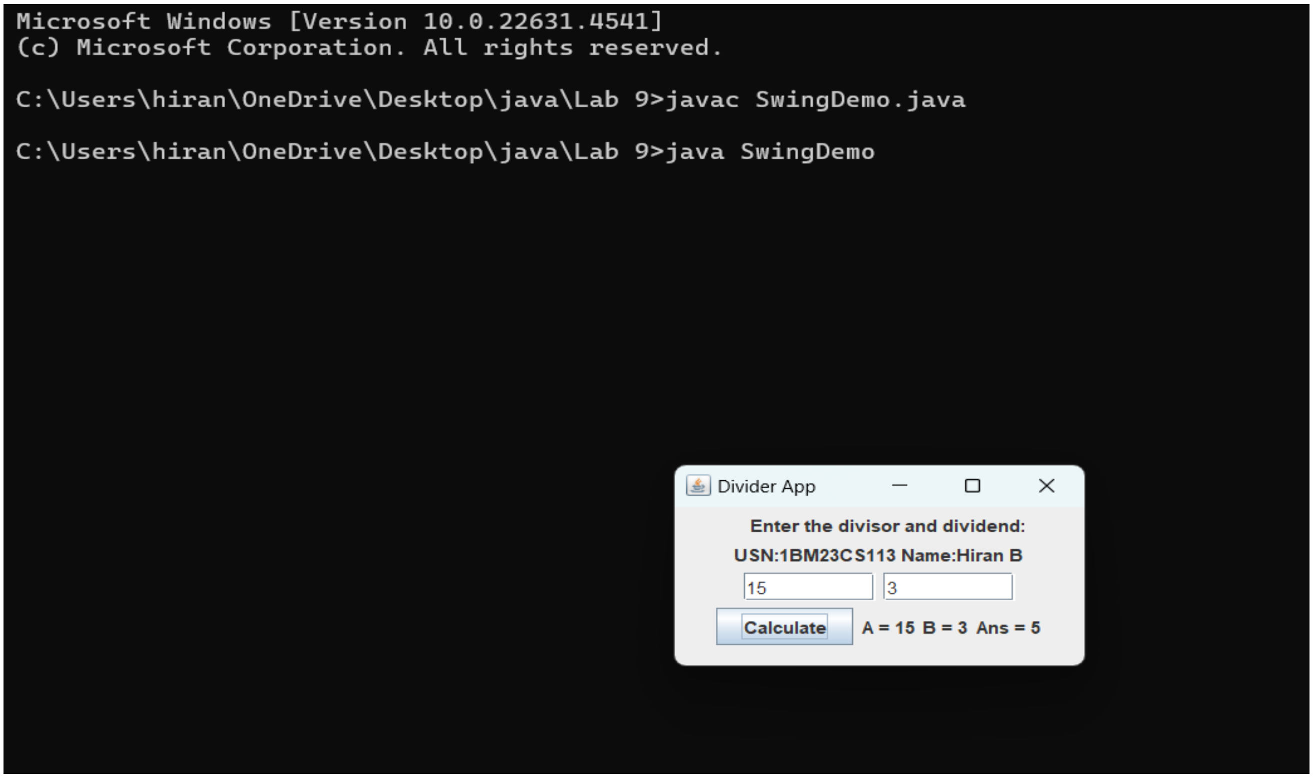
DATE: PAGE: 31

} catch (Arthimetic expression){
JOption Pane . show Message Dialog (frame, " Division by
zero is ~~not~~ not allowed "! "Arthimetic Error" .
JOption Pane. ERROR_MESSAGE);
} catch (Exception ex){
JOption Pane. show Message Dialog (frame.
"An ~~a~~ unexpected error occured : " + ex.
get Message () "Error", JOption Pane. ERROR_MESSAGE);
}
}
});
}
}

Code:

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

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel("Enter the divisor and dividend:");
        JLabel jlab1 = new JLabel("USN:1BM23CS113 Name:Hiran B");
        JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);
JButton button = new JButton("Calculate");
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();

jfrm.add(err);
jfrm.add(jlab);
jfrm.add(jlab1);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

ActionListener l = new ActionListener() {
   public void actionPerformed(ActionEvent evt) {
      System.out.println("Action event from a text field");
   }
};

ajtf.addActionListener(l);
bjtf.addActionListener(l);

button.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent evt) {
      try {
         int a = Integer.parseInt(ajtf.getText());
         int b = Integer.parseInt(bjtf.getText());
         int ans = a / b;

         alab.setText("A = " + a);
         blab.setText("B = " + b);
         anslab.setText("Ans = " + ans);
         err.setText("");
      } catch (NumberFormatException e) {
         alab.setText("");
         blab.setText("");
```

```
                anslab.setText("");
                err.setText("Enter Only Integers!");
            } catch (ArithmeticException e) {
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText("B should be NON zero!");
            }
        }
    });

    jfrm.setVisible(true);
}

public static void main(String args[]) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SwingDemo();
        }
    });
}
}
```



```
Microsoft Windows [Version 10.0.22631.4541]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hiran\OneDrive\Desktop\java\Lab 9>javac SwingDemo.java

C:\Users\hiran\OneDrive\Desktop\java\Lab 9>java SwingDemo
```

Enter the divisor and dividend:

USN:1BM23CS113 Name:Hiran B

15 | 3

A = 15 B = 3 Ans = 5