**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Belgaum -590014, Karnataka.**

****

**LAB REPORT**

**on**

**Object Oriented Java Programming (23CS3PCOOJ)**

***Submitted by***

**Bhoomika M (1BM23CS068)**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

****

**B.M.S. COLLEGE OF ENGINEERING**

**(Autonomous Institution under VTU)**

**BENGALURU-560019**

**Sep-2024 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**

****

**CERTIFICATE**

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **Bhoomika M (1BM23CS068),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. 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.

|  |  |
| --- | --- |
| Prof.Swathi Sridharan  Assistant Professor  Department of CSE, BMSCE | Dr.Kavitha Sooda  Professor & HOD  Department of CSE, BMSCE |

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.**  **No.** | **Date** | **Experiment Title** | **Page No.** |
| 1 | 01/10/2024 | Roots of Quadratic Equations | 4-5 |
| 2 | 08/10/2024 | SGPA of a Student | 6-9 |
| 3 | 15/10/2024 | Book Information | 9-12 |
| 4 | 22/10/2024 | Abstract Class-Animal and Shape | 13-17 |
| 5 | 29/10/2024 | Bank Class | 17-23 |
| 6 | 12/11/2024 | Packages | 23-28 |
| 7 | 19/11/2024 | Interfaces | 29-31 |
| 8 | 26/11/204 | Exception Handling | 32-34 |
| 9 | 03/12/2024 | Thread Programming | 34-37 |
| 10 | 03/12/2024 | Open Ended Exercise | 37 |

Github Link:

https://github.com/Bhoomika-M-CSE/ooj

**Program 1**

Implement Quadratic Equation

Code:

import java.util.Scanner;

public class QuadraticEquationSolver {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the coefficients of the quadratic equation (a, b, c):");

double a = scanner.nextDouble();

double b = scanner.nextDouble();

double c = scanner.nextDouble();

if (a == 0) {

System.out.println("This is not a quadratic equation.");

} else {

double discriminant = b \* b - 4 \* a \* c;

if (discriminant > 0) {

// Two distinct real roots

double root1 = (-b + Math.sqrt(discriminant)) / (2 \* a);

double root2 = (-b - Math.sqrt(discriminant)) / (2 \* a);

System.out.println("The roots are real and distinct.");

System.out.println("Root 1: " + root1);

System.out.println("Root 2: " + root2);

}

else if (discriminant == 0){

double root = -b / (2 \* a);

System.out.println("The roots are real and equal.");

System.out.println("Root: " + root);

}

else {

double realPart = -b / (2 \* a);

double imaginaryPart = Math.sqrt(-discriminant) / (2 \* a);

System.out.println("The roots are complex and distinct.");

System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");

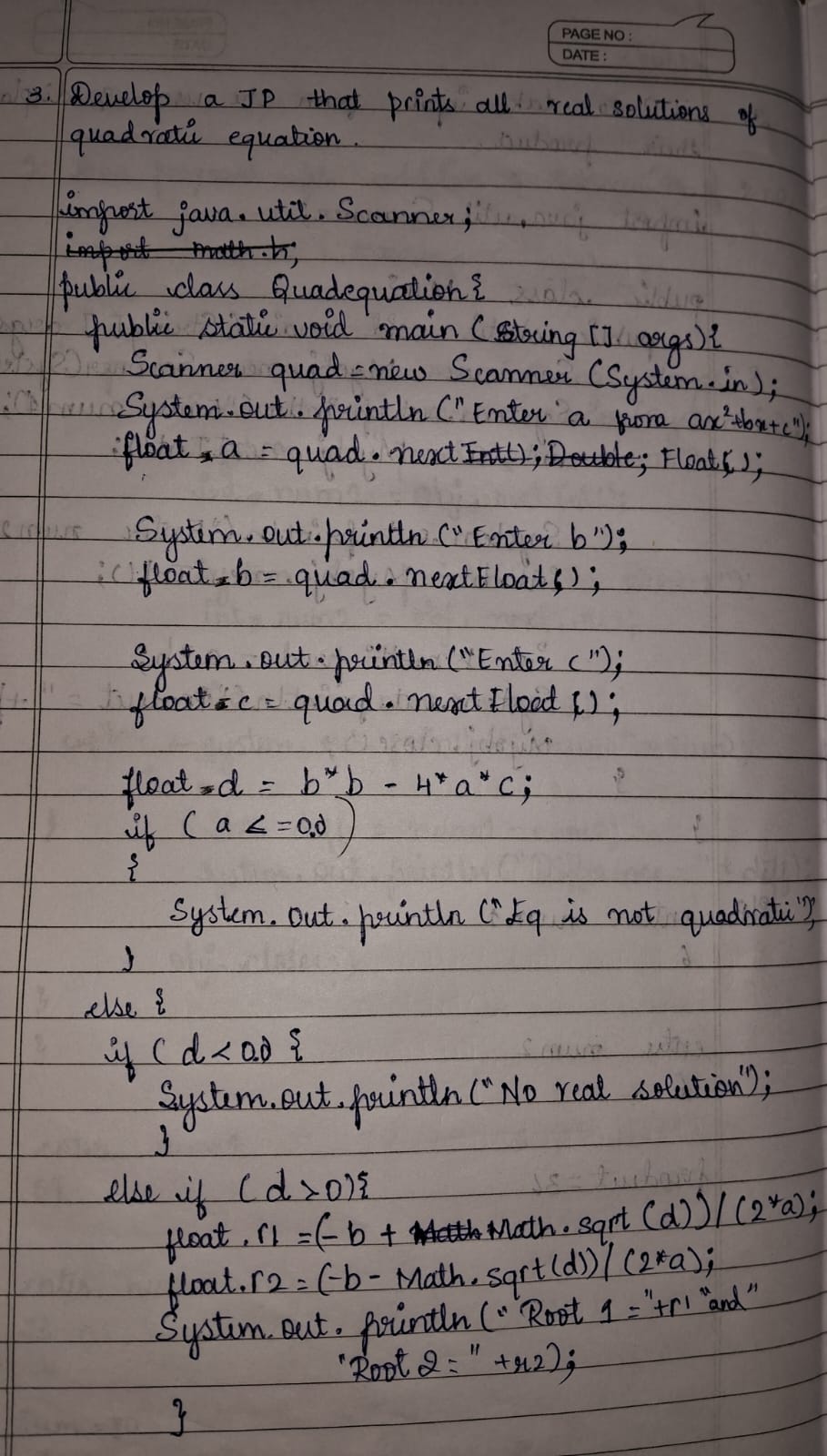
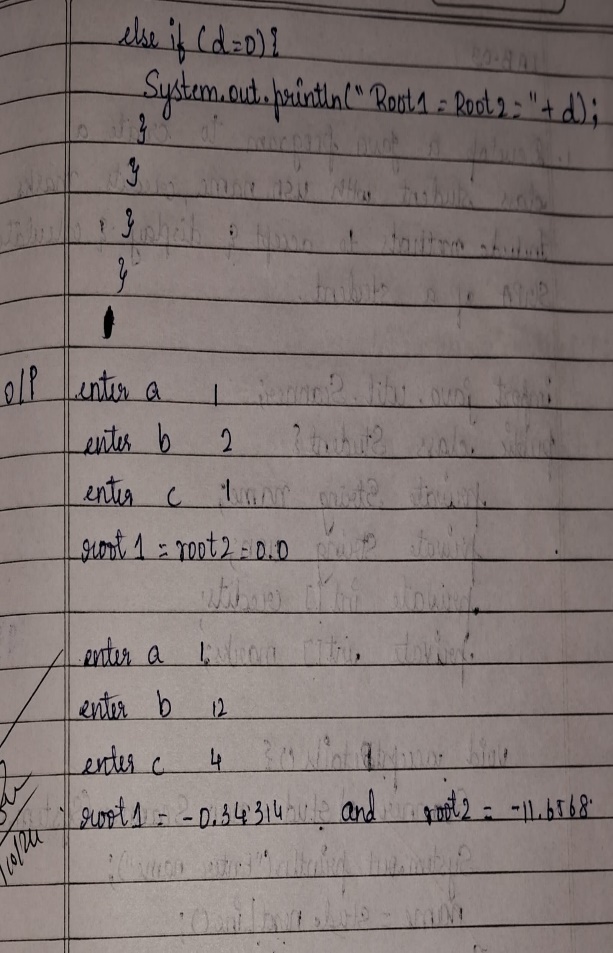
System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");

}

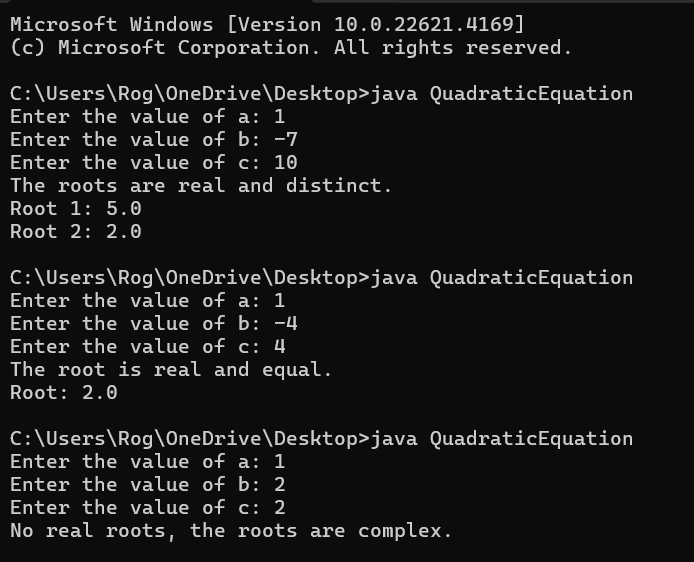
}

}

}

Output:



Program 2

SGPA of a Student

Code:

import java.util.Scanner;

class Subject {

int subM;

int cred;

int grade;

void setSubDet(int marks, int cred) {

this.subM = marks;

this.cred = cred;

if (subM >= 90) {

grade = 10; }

else if (subM >= 80) {

grade = 9; }

else if (subM >= 70) {

grade = 8; }

else if (subM >= 60) {

grade = 7; }

else if (subM >= 50) {

grade = 6; }

else if (subM >= 40) {

grade = 5; }

else {

grade = 0;}

}}

class Student {

String name;

String usn;

double SGPA;

Scanner s = new Scanner(System.in);

Subject[] subjects = new Subject[8];

Student() {

for (int i = 0; i < subjects.length; i++) {

subjects[i] = new Subject(); }

}

void getMarks() {

for (int i = 0; i < subjects.length; i++) {

System.out.print("Enter marks for subject " + (i + 1) + ": ");

int marks = s.nextInt();

System.out.print("Enter credit for subject " + (i + 1) + ": ");

int cred = s.nextInt();

subjects[i].setSubDet(marks, cred); }

}

double calSGPA() {

double Score = 0;

int totalCred = 0;

for (Subject subject : subjects) {

Score += (subject.grade \* subject.cred);

totalCred += subject.cred; }

if (totalCred > 0) {

SGPA = Score / totalCred; }

else {

SGPA = 0; }

return SGPA;

}

public class StudentDetails {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

int numSems = sc.nextInt();

Student[] students = new Student[ numSems];

double c=0.0;

String usn,name;

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

usn = sc.next();

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

name = sc.next();

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

System.out.println("Enter details for semester" + (i + 1));

students[i] = new Student();

students[i].getMarks();

double s=students[i].calSGPA();

c+=s;

}

c=c/numSems;

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

System.out.println("USN: " + usn);

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

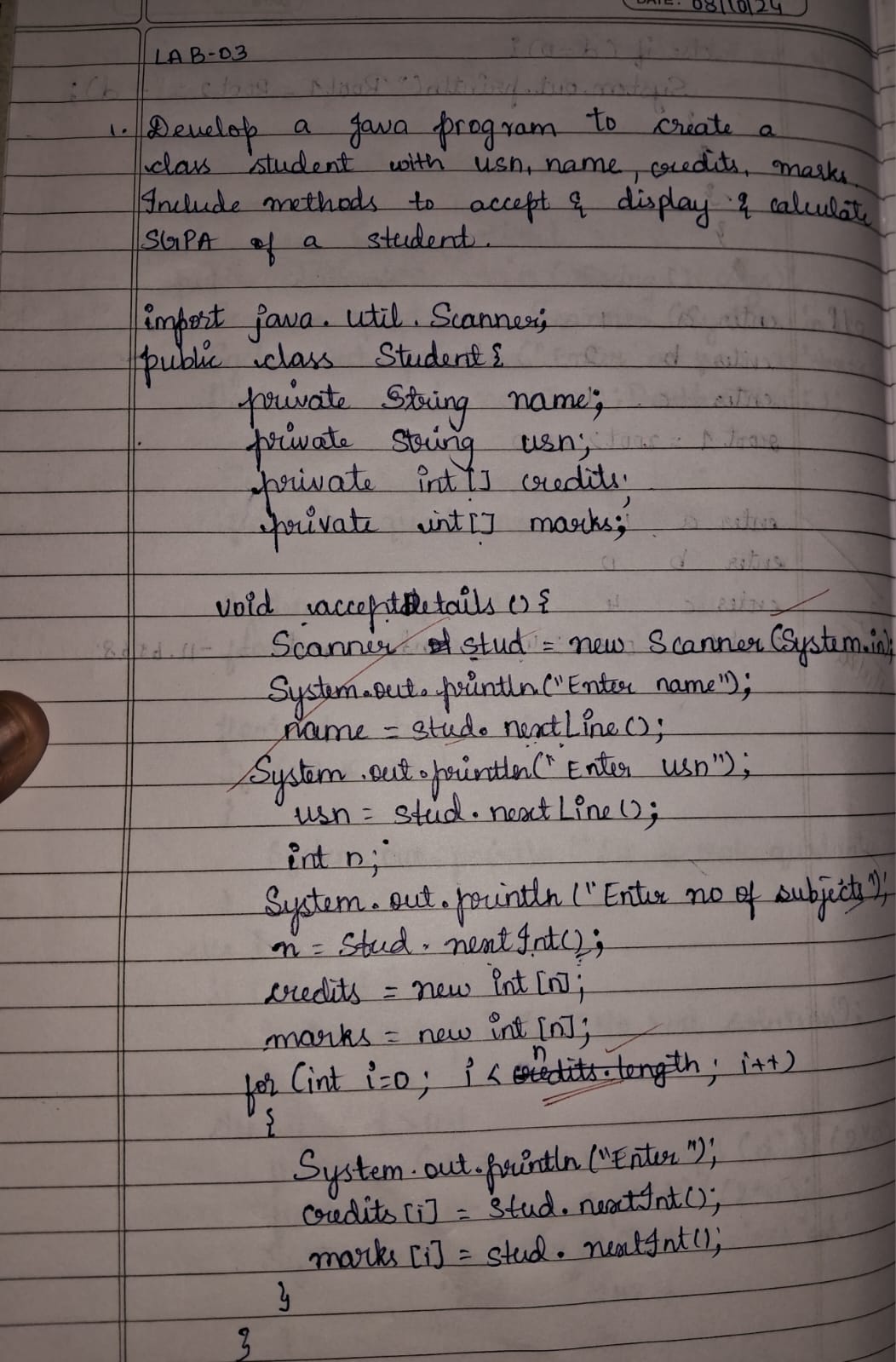
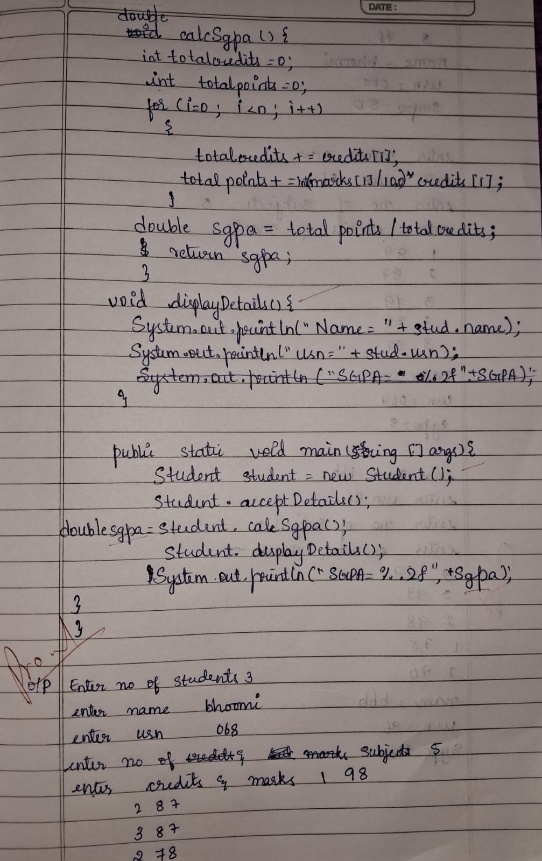
System.out.println("SGPA for sem "+ (i+1)+":" + students[i].calSGPA());}

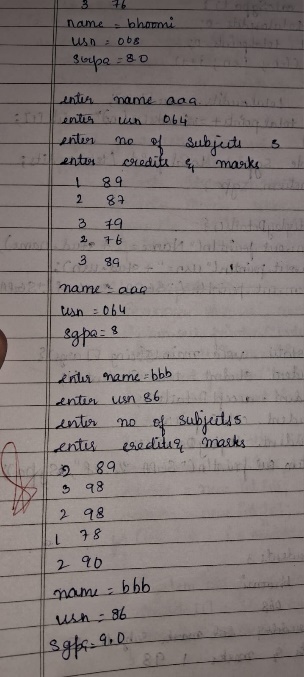
System.out.println("CGPA: " +c);

}

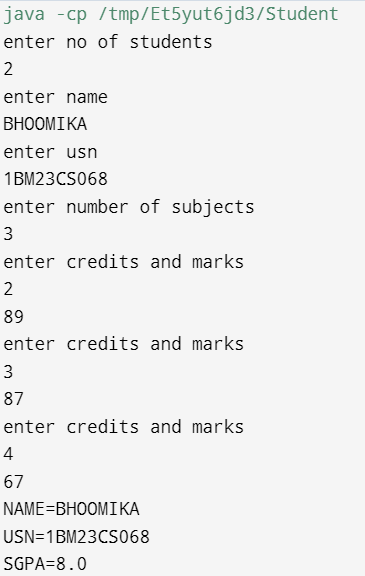
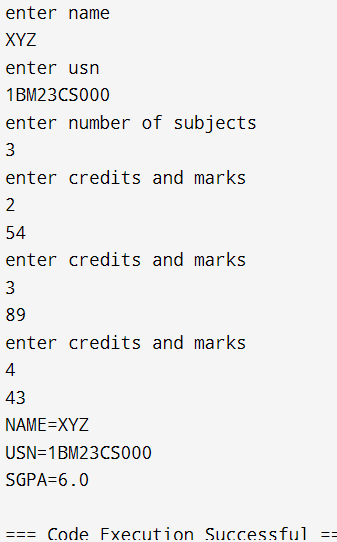
}

}



Output:

Program 3

Book Information

Code:

import java.util.Scanner;

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 void setAuthor(String author) {

this.author = author;

}

public void setPrice(double price) {

this.price = price;

}

public void setNumPages(int num\_pages) {

this.num\_pages = num\_pages;

}

public String getName() {

return name;

}

public String getAuthor() {

return author;

}

public double getPrice() {

return price;

}

public int getNumPages() {

return num\_pages;

}

public String toString() {

return "Book name: " + name + "\n" +

"Author: " + author + "\n" +

"Price: $" + price + "\n" +

"Number of pages: " + num\_pages + "\n";

}

}

class Main {

public static void main(String[] args) {

Scanner ob = new Scanner(System.in);

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

int n = ob.nextInt();

ob.nextLine();

Book[] books = new Book[n];

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

System.out.println("Enter name of the book " + (i + 1) + ":");

String name = ob.nextLine();

System.out.println("Enter name of the author:");

String author = ob.nextLine();

System.out.println("Enter price of the book:");

double price = ob.nextDouble();

System.out.println("Enter number of pages:");

int num\_pages = ob.nextInt();

ob.nextLine();

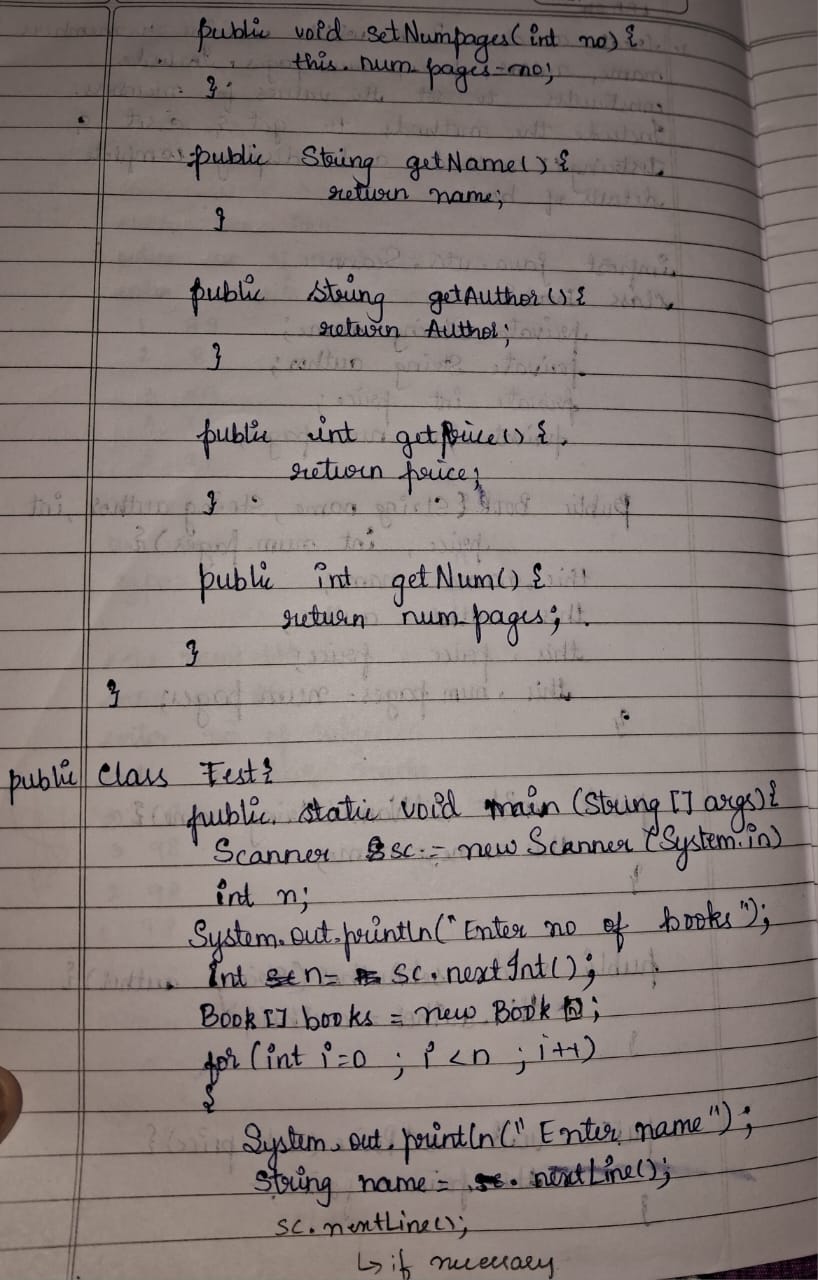
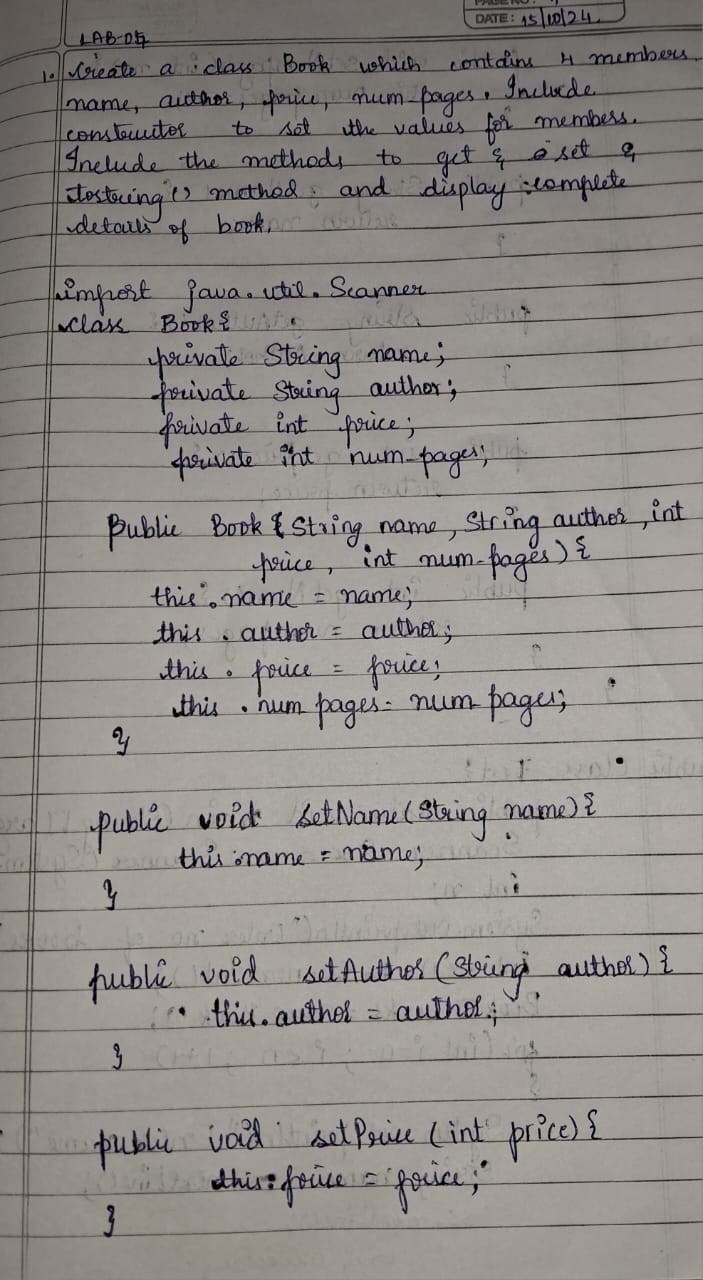
books[i] = new Book(name, author, price, num\_pages);

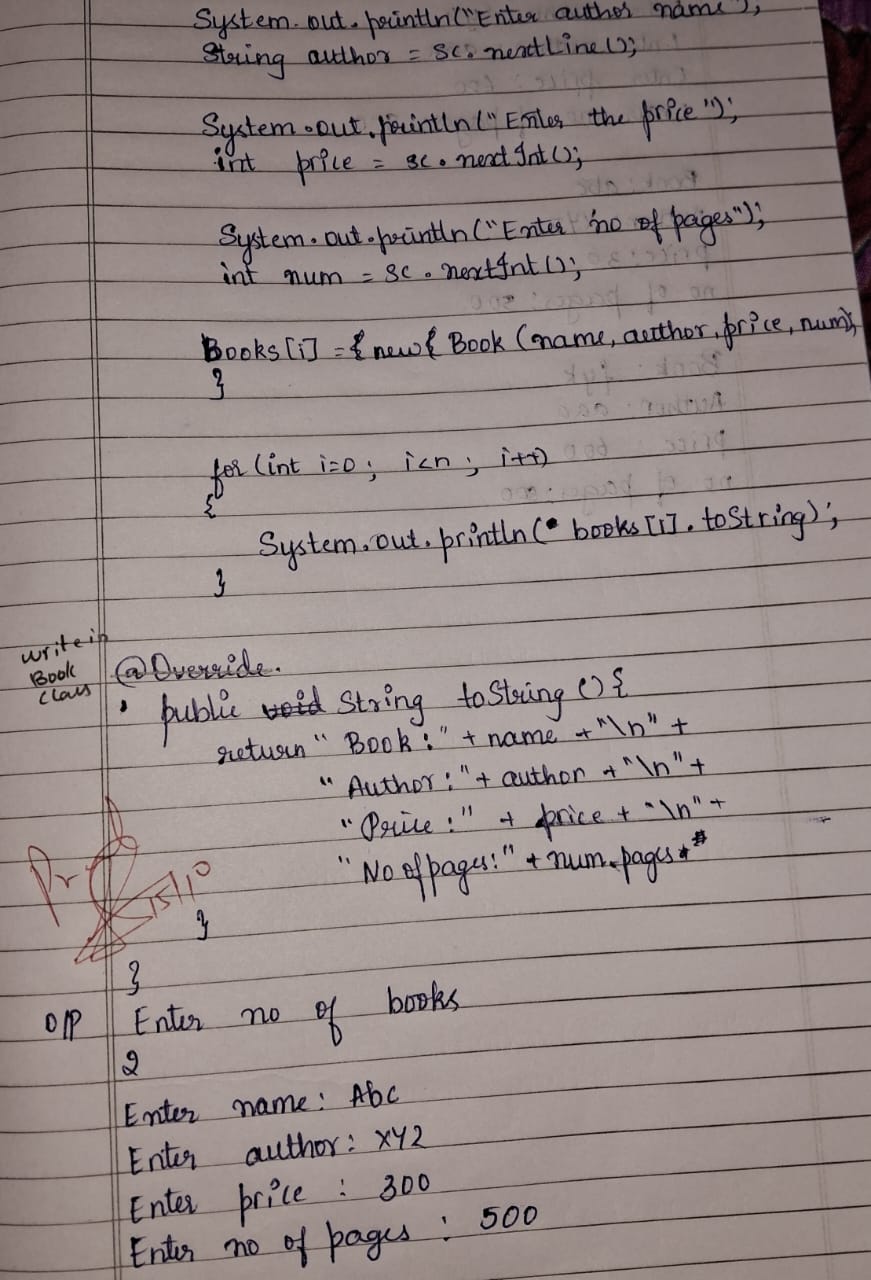
System.out.println(books[i].toString());

}

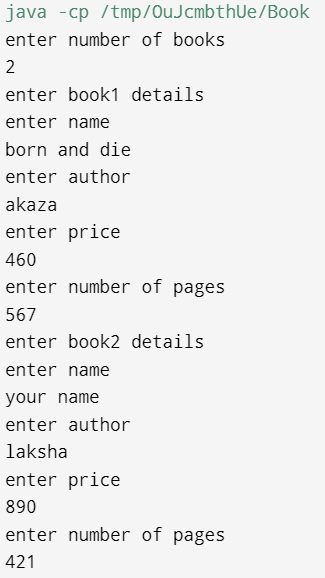
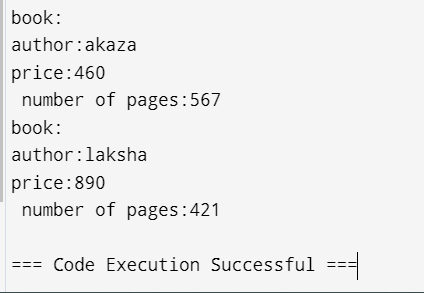
}

}



Output:

Program 4

Abstract Class

Animal Class:

Code:

abstract class Animal {

abstract void eatAndSleep();

}

class Lion extends Animal {

void eatAndSleep() {

System.out.println("Lion: Hunts for prey, eats meat, and sleeps in a den.");

}

}

class Deer extends Animal {

void eatAndSleep() {

System.out.println("Deer: Grazes on grass, drinks water, and sleeps under trees.");

}

}

class Tiger extends Animal {

void eatAndSleep() {

System.out.println("Tiger: Stalks its prey, eats meat, and rests in dense forests.");

}

}

public class Main {

public static void main(String[] args) {

Animal lion = new Lion();

Animal deer = new Deer();

Animal tiger = new Tiger();

System.out.println("Animal Behaviors:");

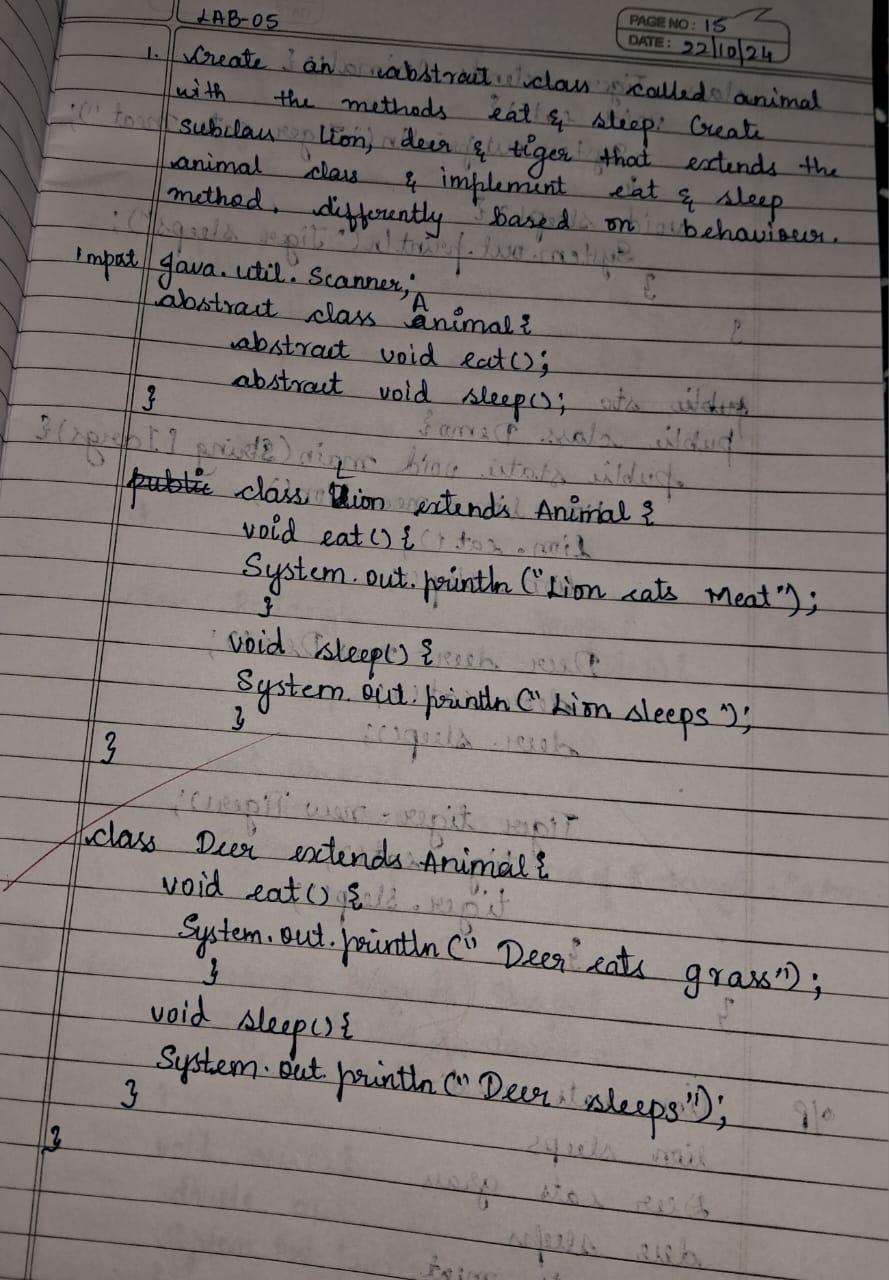
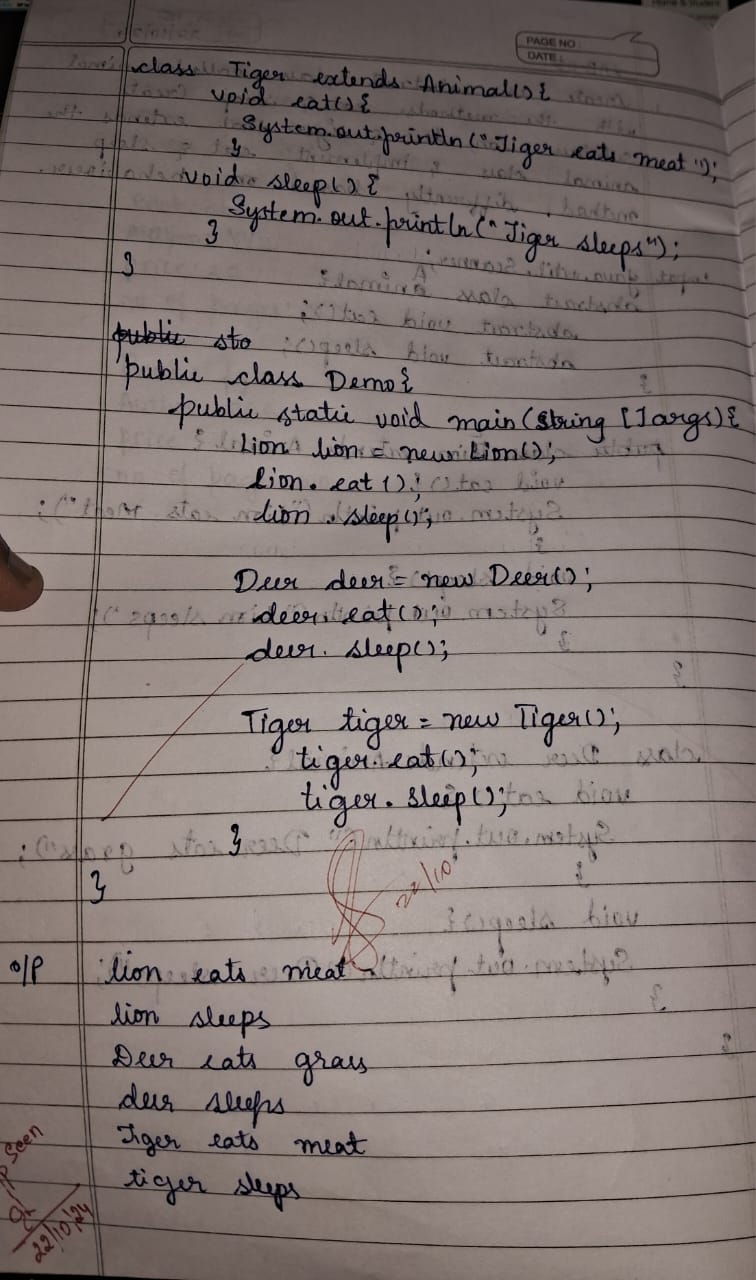
lion.eatAndSleep();

deer.eatAndSleep();

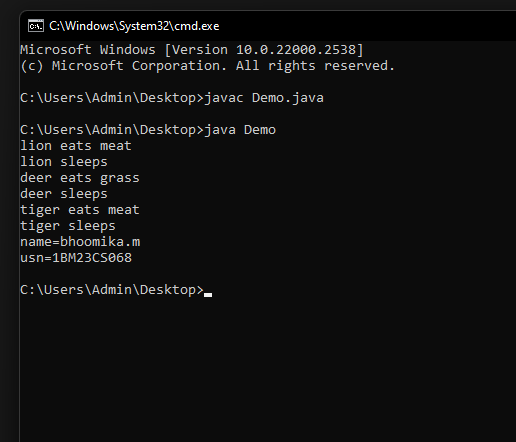
tiger.eatAndSleep();

    }

}

Output:



Shape class:

Code:

import java.util.Scanner;

interface Polygon{

default double getPerimeter(){

Scanner sc=new Scanner(System.in);

System.out.println("Enter number of sides:");

int n=sc.nextInt();

double p=0;

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

System.out.println("Enter side:");

double a=sc.nextDouble();

p+=a;

}

return p;

}

abstract double getArea();

}

class Rectangle implements Polygon{

public double length;

public double breadth;

public Rectangle(double length,double breadth){

this.length=length;

this.breadth=breadth;}

public double getArea(){

double area=length\*breadth;

return area;

}

}

class Triangle implements Polygon{

public double length1;

public double breadth1;

public Triangle(double length1,double breadth1){

this.length1=length1;

this.breadth1=breadth1;}

public double getArea(){

double area=0.5\*length1\*breadth1;

return area;

}

}

public class Shape{

public static void main(String []Args){

Scanner sc=new Scanner(System.in);

System.out.println("Bhoomika BG-1BM23CS067");

do{

System.out.println("Choose:\n 1.Rectangle\n 2.Triangle");

int x=sc.nextInt();

switch(x){

case(1):

{

System.out.println("Enter length:");

double l=sc.nextDouble();

System.out.println("Enter breadth:");

double b=sc.nextDouble();

Polygon rc=new Rectangle(l,b);

double p=rc.getPerimeter();

System.out.println("Perimeter of rectangle is:"+p);

double a=rc.getArea();

System.out.println("Area of rectangle is:"+a);

break;

}

case(2):{

System.out.println("Enter base:");

double l=sc.nextDouble();

System.out.println("Enter height:");

double b=sc.nextDouble();

Polygon tr=new Triangle(l,b);

double p1=tr.getPerimeter();

System.out.println("Perimeter of triangle is:"+p1);

double a1=tr.getArea();

System.out.println("Area of triangle is:"+a1);

break;

}

default:

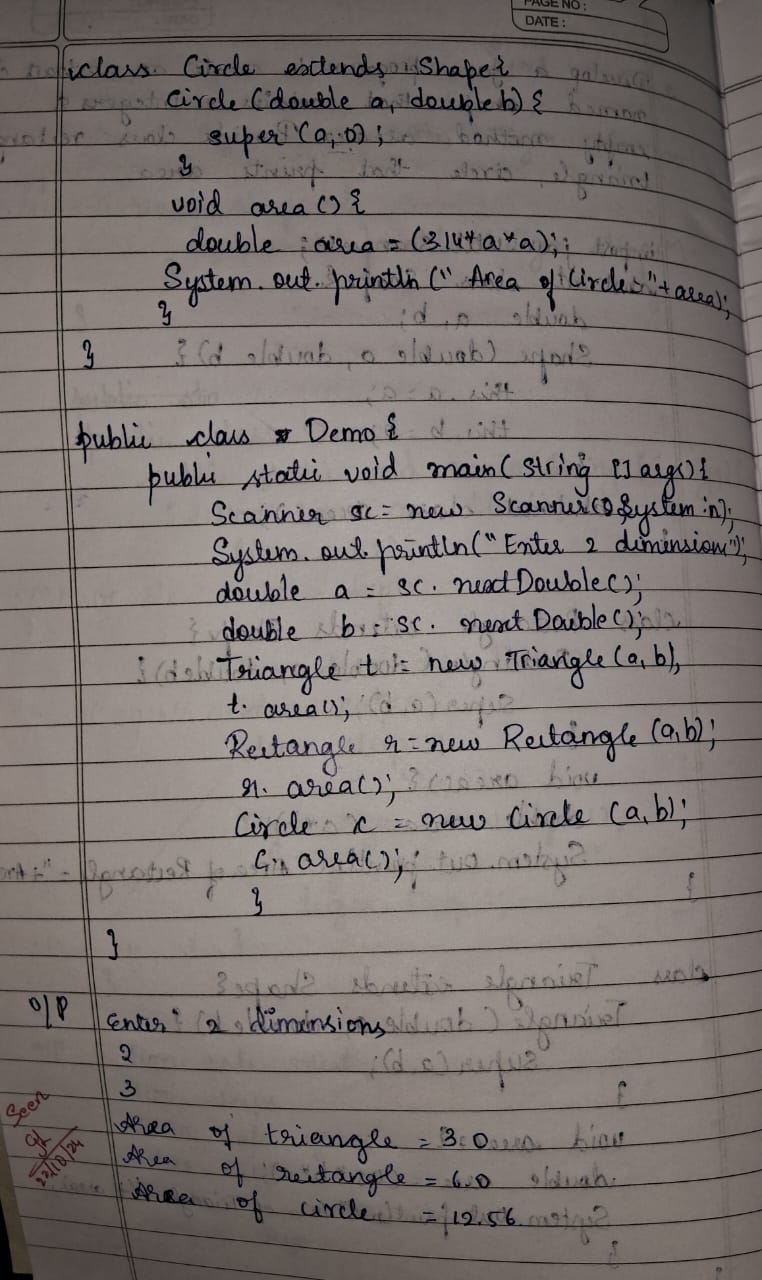
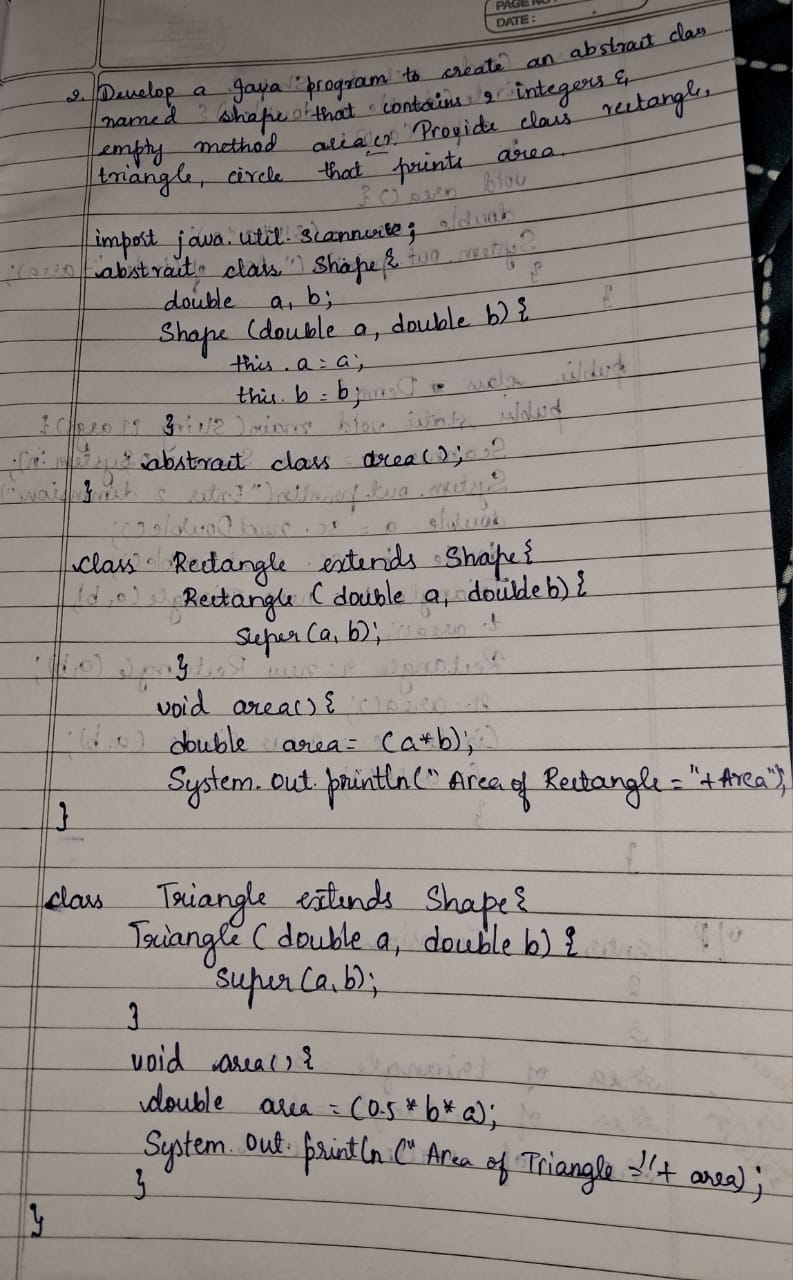
System.exit(0);

}

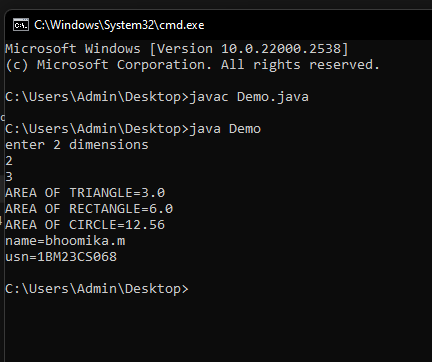
}while(true);

}

}

Output:



Program 5

Bank Class

Code:

import java.util.Scanner;

class Account {

private String customerName;

private int accountNumber;

private double balance;

public Account(String customer\_name, int acc\_no, double balance) {

this.customer\_name = customer\_name;

this.acc\_no = acc\_no;

this.balance = balance;

}

public double getBalance() {

return balance;

}

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

System.out.println("Deposited: " + amount);

System.out.println("Your new balance is:"+balance);

} else {

System.out.println("Deposit amount must be positive.");

}

}

public void displayBalance() {

System.out.println("Current Balance: " + balance);

}

}

class SavingsAccount extends Account {

private double interestRate;

public SavingsAccount(String customerName, int accountNumber, double initialBalance, double interestRate) {

super(customerName, accountNumber, initialBalance);

this.interestRate = interestRate;

}

public void computeAndDepositInterest() {

double interest = getBalance() \* interestRate / 100;

deposit(interest);

System.out.println("Balance is: "+balance);

System.out.println("Interest of " + interest + " has been credited.");

}

public void withdraw(double amount)

{

if(amount<=getBalance()){

double new\_balance=getBalance()-amount;

System.out.println("withdrew:"+amount + " balance is:"+ new\_balance);

}

else

System.out.println("Insufficient funds!!");

}

}

class CurrentAccount extends Account {

private double minimumBalance;

private double serviceCharge;

public CurrentAccount(String customerName, int accountNumber, double initialBalance, double minimumBalance, double serviceCharge) {

super(customerName, accountNumber, initialBalance);

this.minimumBalance = minimumBalance;

this.serviceCharge = serviceCharge;

}

public void withdraw(double amount) {

if (amount <= getBalance()) {

double newBalance = getBalance() - amount;

System.out.println("Withdrew: " + amount);

} else {

System.out.println("Insufficient balance.");

}

}

private void checkMinimumBalance() {

if (getBalance() < minimumBalance) {

System.out.println("Balance is below minimum");

balance-=serviceCharge;

System.out.println("Deducted service charge:" +serviceCharge);

System.out.println("Balance after deduction is":+balance);

}

}

}

public class Bank {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("enter customer name:");

String name=sc.nextLine();

System.out.println("enter accno:");

int acc\_no=sc.nextInt();

System.out.println("enter initial balance:");

double balance=sc.nextDouble();

System.out.println("enter minimum balance:");

double minimum\_balance=sc.nextDouble();

System.out.println("enter interest rate:");

double interest\_rate=sc.nextDouble();

System.out.println("Enter choice:\n 1.Current acc\n 2.Savings acc");

int ch=sc.nextInt();

switch(ch){

case(1):

System.out.println("account is current type");

CurrentAccount cu=new CurrentAccount(name,acc\_no,balance,minimum\_balance,interest\_rate);

do{ System.out.println("enter choice:\n 1.deposit\n 2.withdraw\n 3.display balance");

int c=sc.nextInt();

cu.checkMinimumBalance()

if(c==1){

System.out.println("enter amount to be deposited:");

double amt=sc.nextDouble();

cu.deposit(amt);}

else if(c==2){

System.out.println("enter amount to withdraw:");

double amt=sc.nextDouble();

cu.withdraw(amt);}

else if(c==3){

cu.displayBalance();}

else

System.out.println("invalid entry!!");

exit(0);

}while(true);

case(2):

System.out.println("account is savings type");

SavingsAccount sa=new SavingsAccount(name,acc\_no,balance,interest\_rate);

do{ System.out.println("enter choice:\n 1.deposit\n 2.withdraw\n 3.display balance");

int c1=sc.nextInt();

if(c1==1){

System.out.println("enter amount to be deposited:");

double amt=sc.nextDouble();

sa.deposit(amt);}

else if(c1==2){

System.out.println("enter amount to withdraw:");

double amt=sc.nextDouble();

sa.withdraw(amt);}

else if(c1==3){

sa.displayBalance();}

else{

System.out.println("invalid entry!!");

exit(0);}

sa.computeAndDepositInterest();

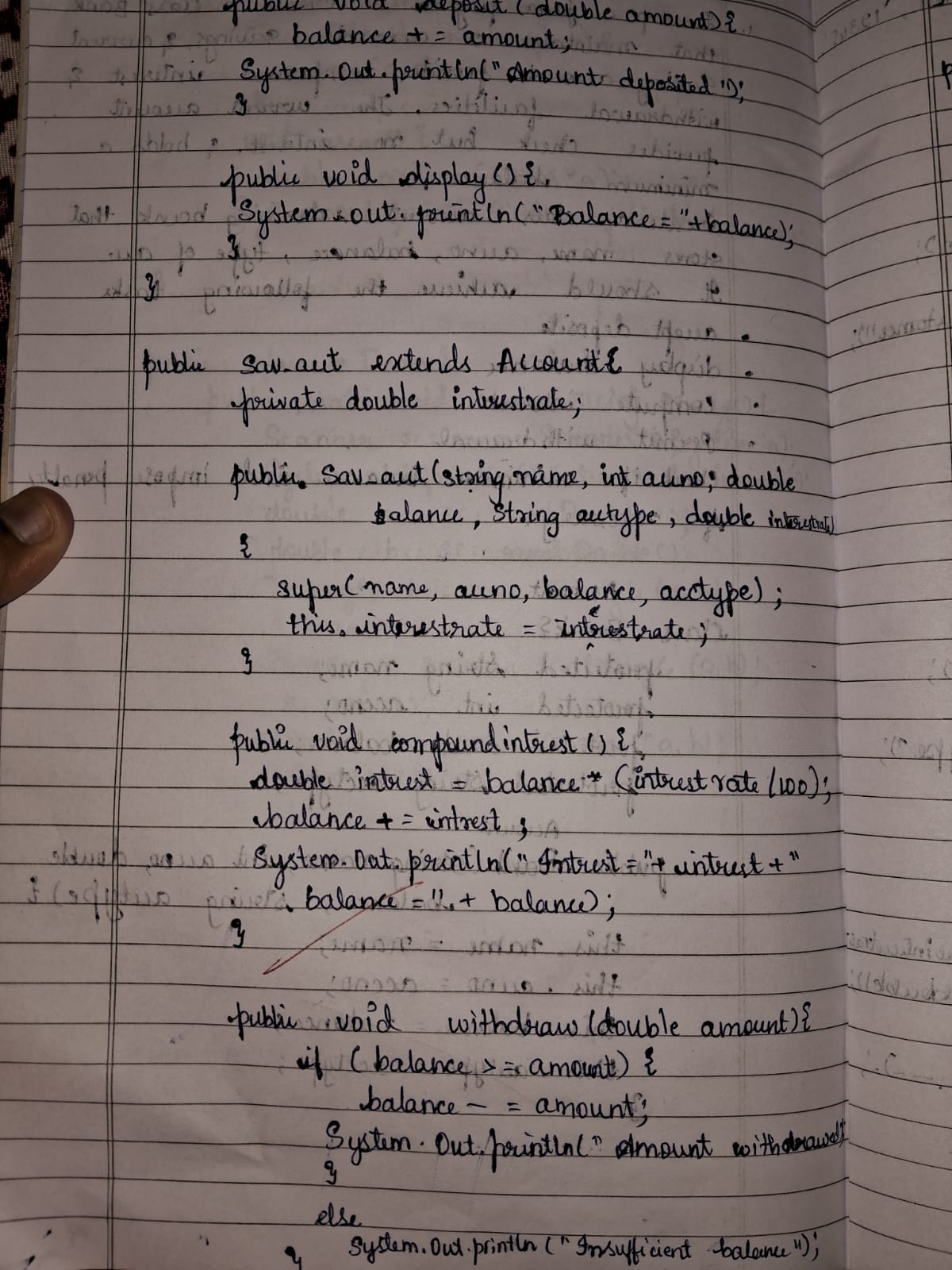
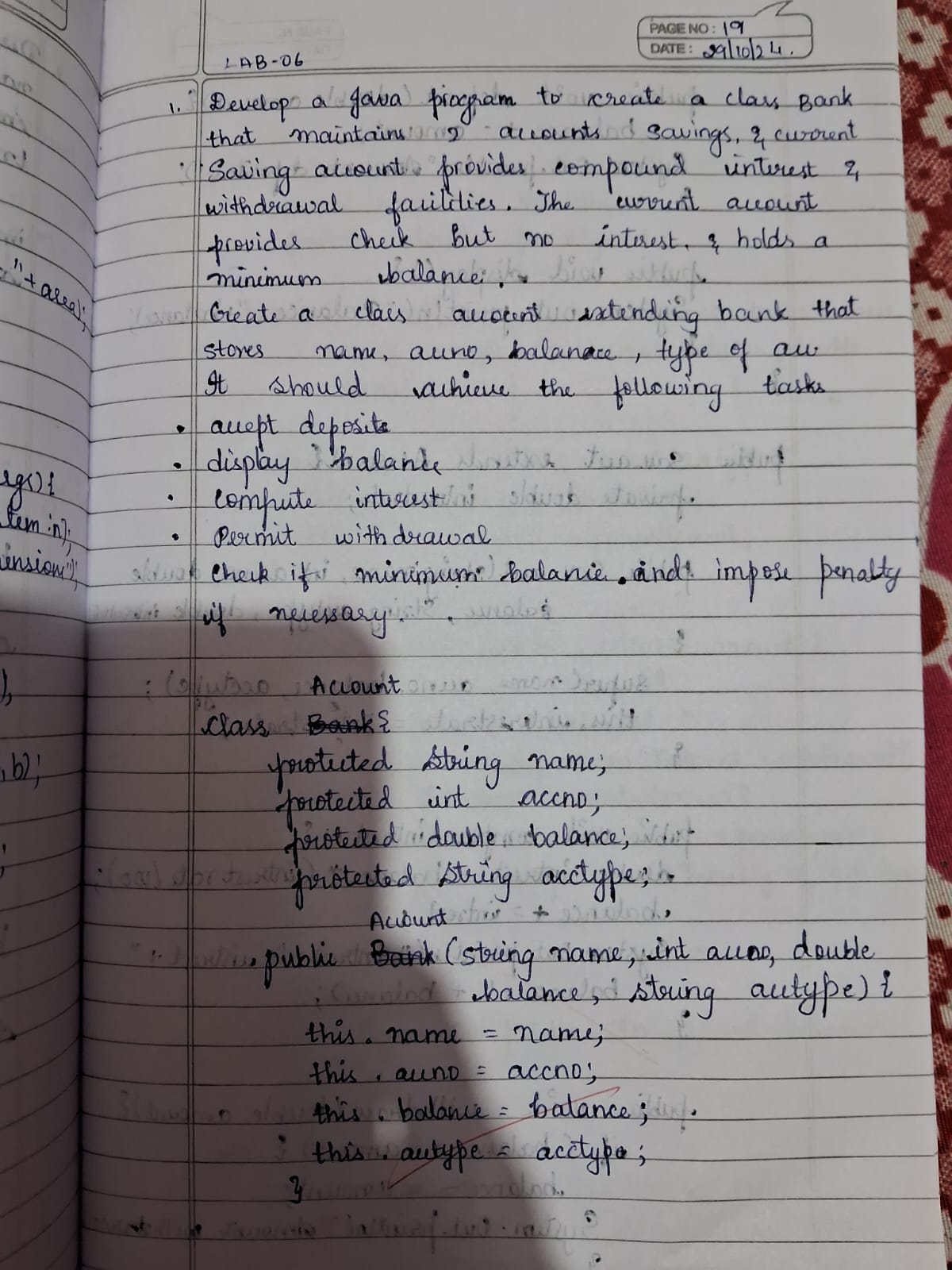
}while(true);

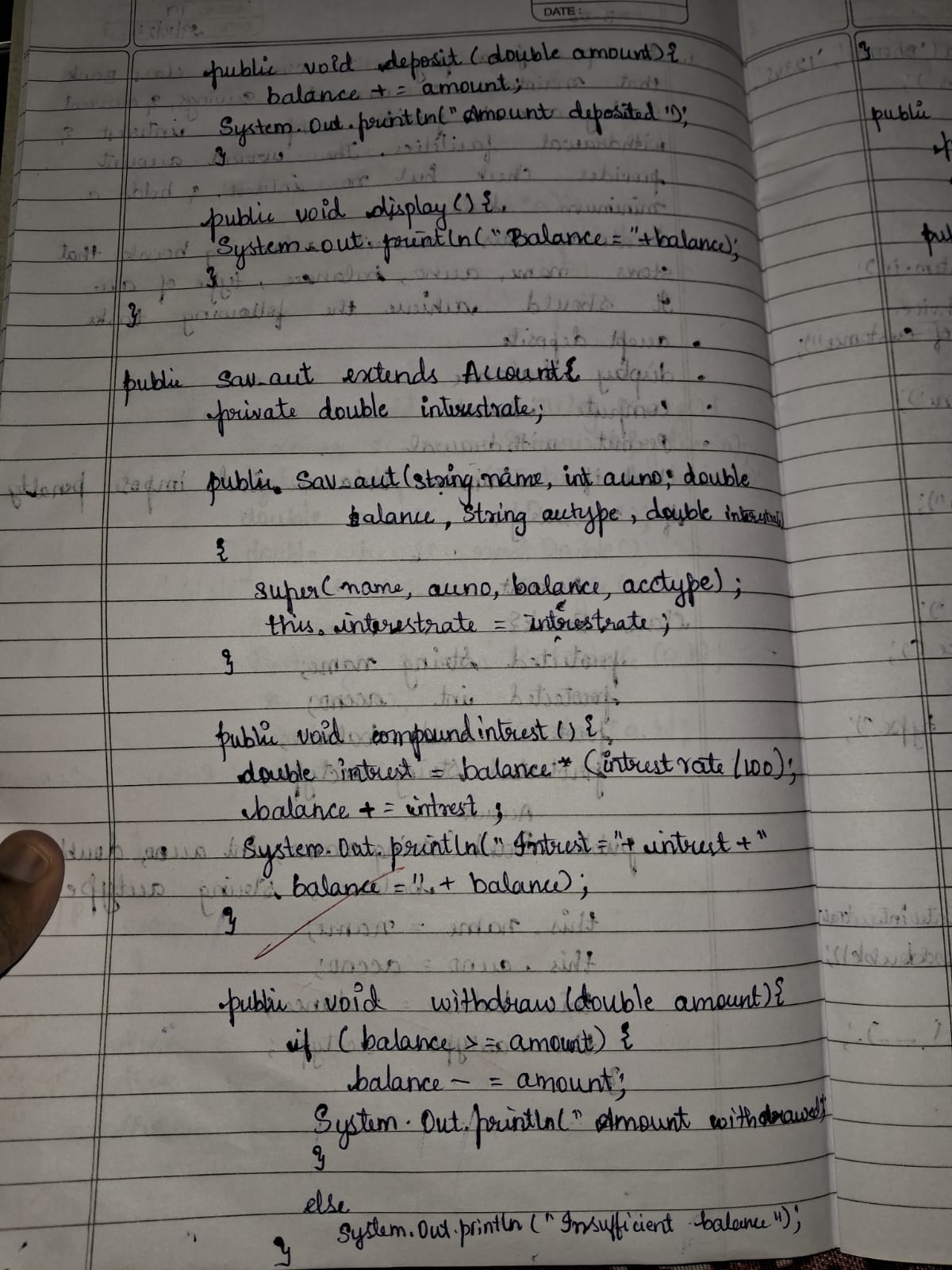
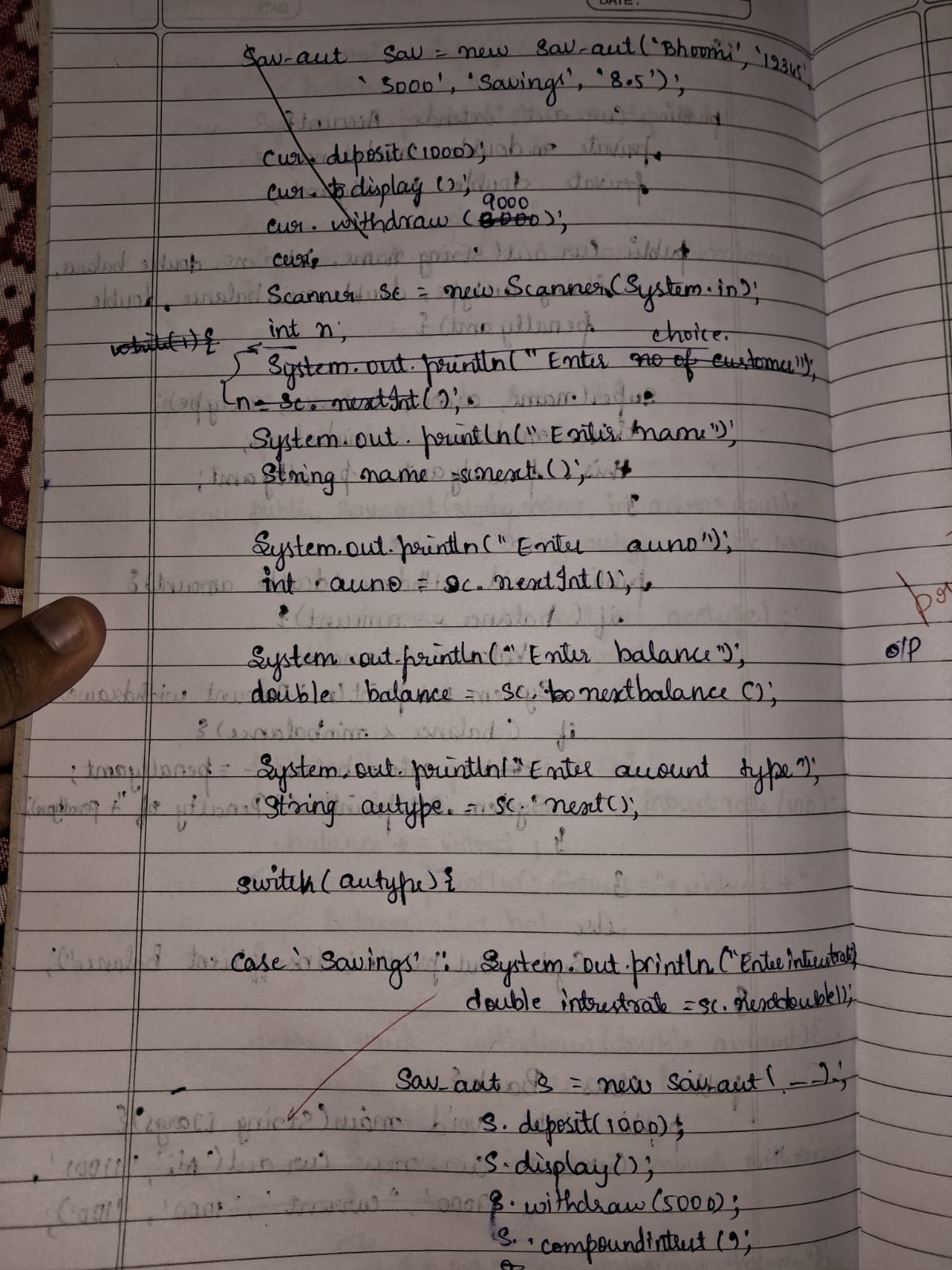
sc.close();

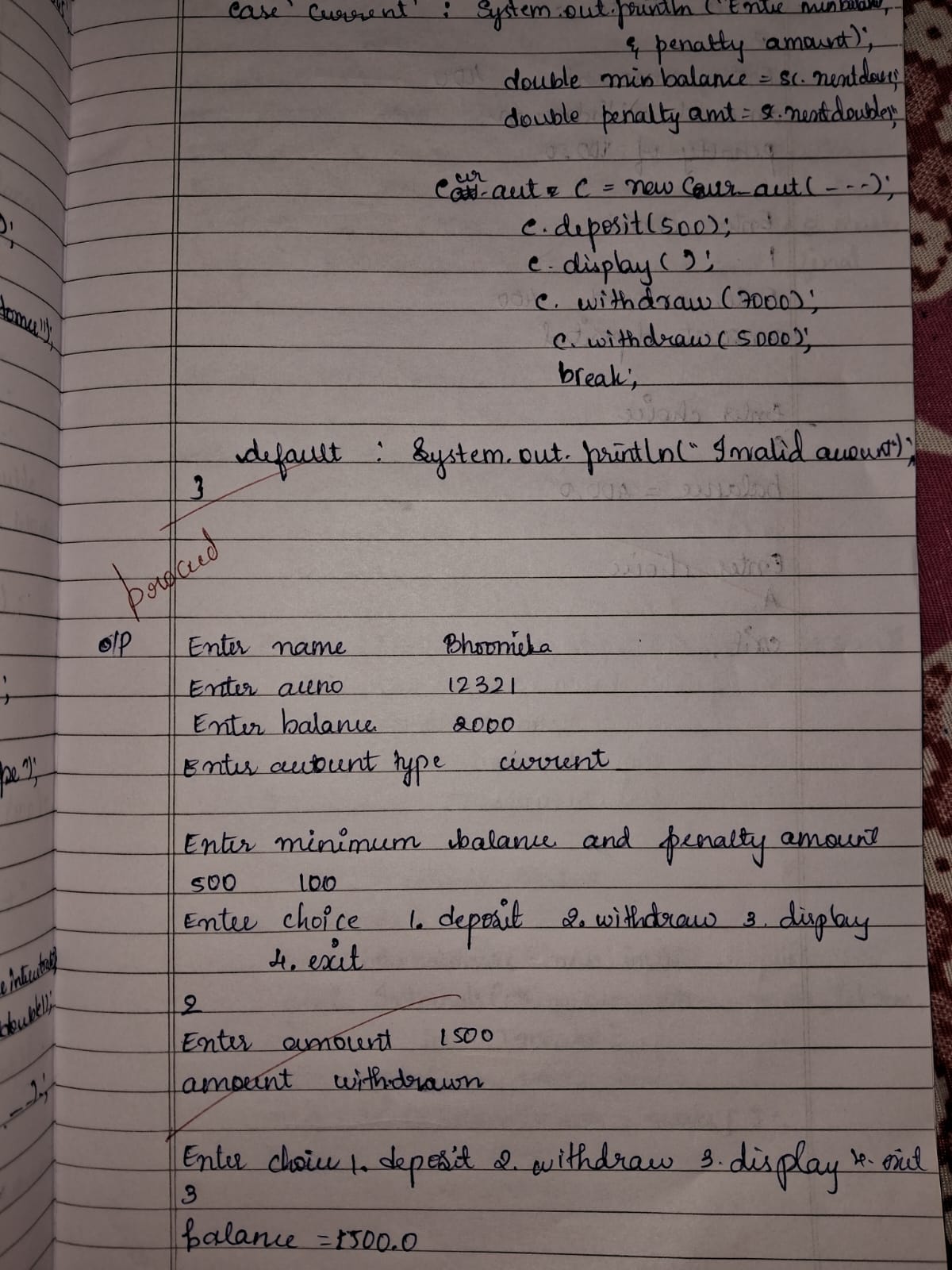
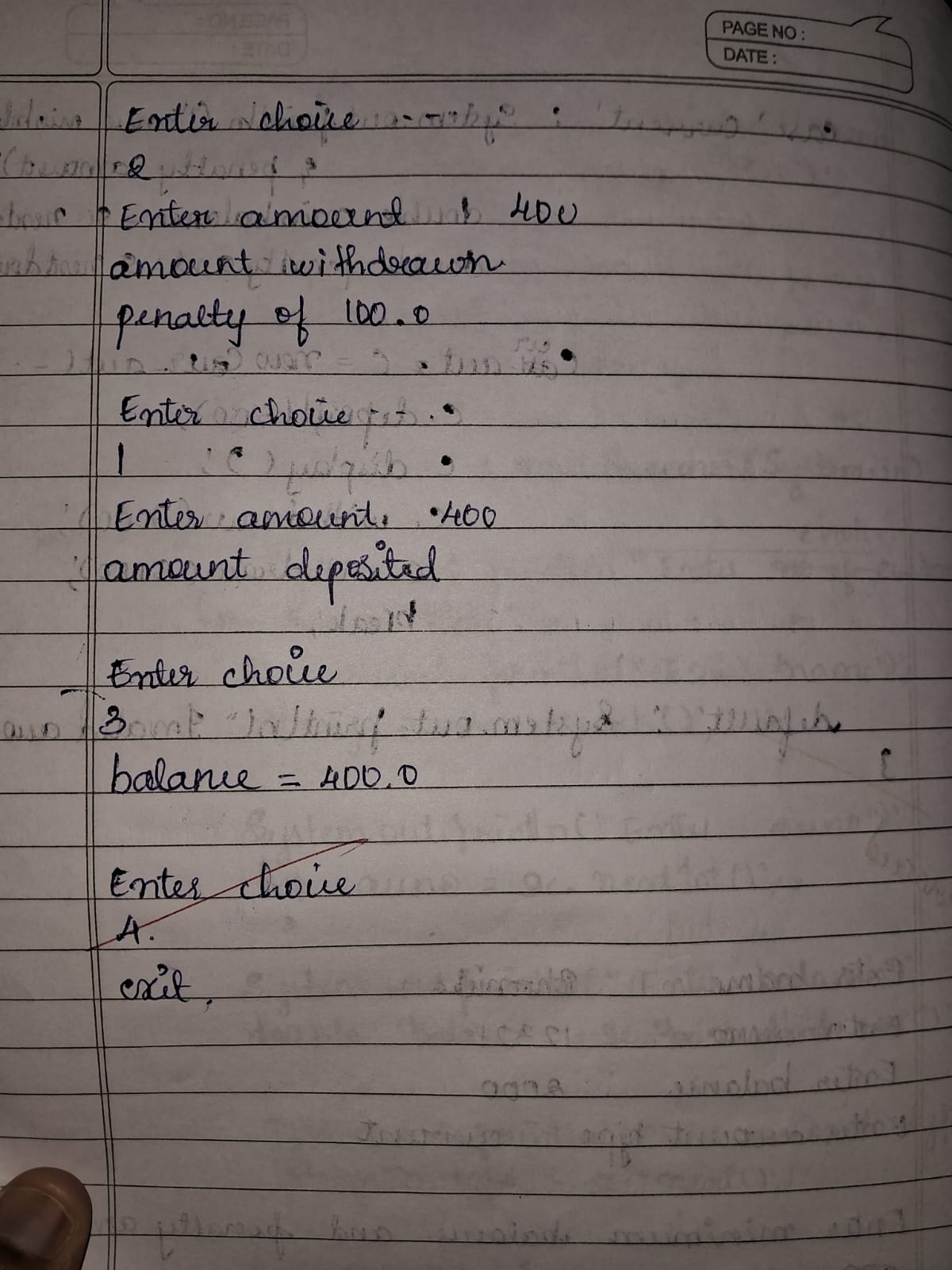
}

}

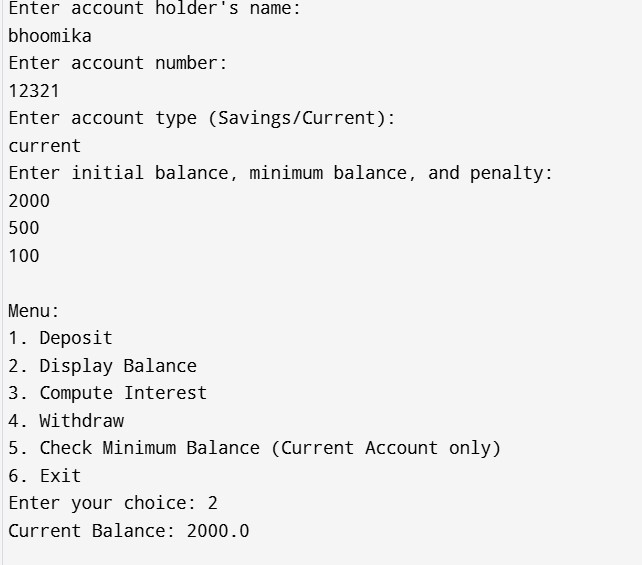
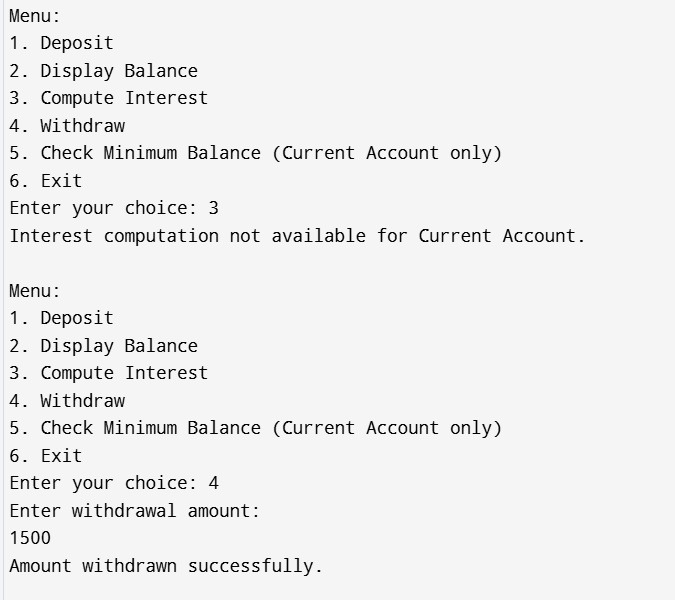
}

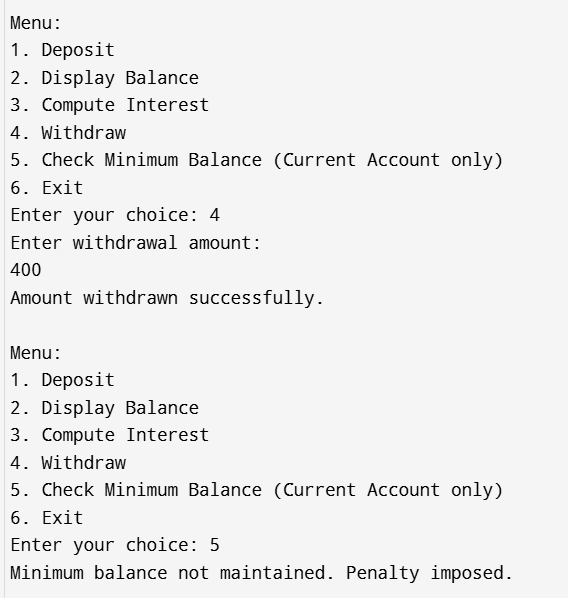
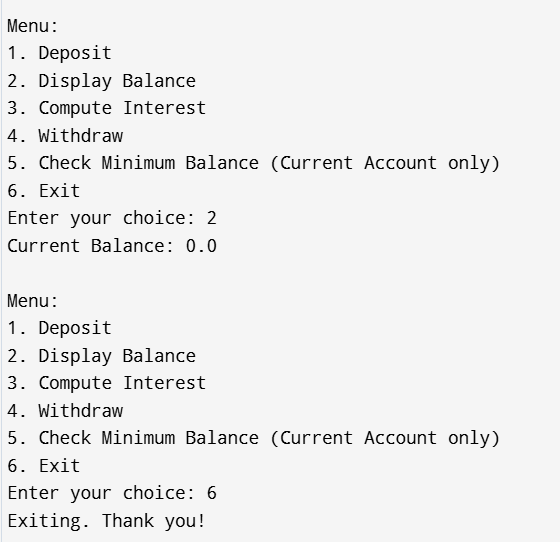


Output:

Program 6

Packages

Code:

package CIE;

public class Student {

public String usn;

public String name;

public int sem;

public Student(String usn, String name, int sem) {

this.usn = usn;

this.name = name;

this.sem = sem;

}

public void displayDetails() {

System.out.println("USN: " + usn);

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

System.out.println("Semester: " + sem);

}

}

public class Internals {

public int[] internalMarks;

public Internals(int[] marks) {

if (marks.length != 5) {

System.out.println("Error: Enter 5 marks!");

return;

}

this.internalMarks = marks;

}

public void displayIMarks() {

System.out.println("Internal Marks: ");

for (int mark : internalMarks) {

System.out.print(mark + " ");

}

System.out.println();

}

}

package SEE;

import CIE.Student;

import CIE.Internals;

public class Externals extends Internals {

public int[] externalMarks;

public Externals(String name, String usn, int sem, int[] marks) {

super(marks);

if (marks.length != 5) {

System.out.println("Error: Enter 5 marks!");

return;

}

this.externalMarks = marks;

}

public void displayEMarks() {

System.out.println("SEE Marks: ");

for (int mark : externalMarks) {

System.out.print(mark + " "); }

System.out.println();

}

}

import CIE.\*;

import SEE.\*;

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 n = sc.nextInt();

sc.nextLine();

Externals[] students = new Externals[n];

Internals[] intMarks = new Internals[n];

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

System.out.println("Enter details for student " + (i + 1) + ":");

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

String usn = sc.nextLine();

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

String name = sc.nextLine();

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

int sem = sc.nextInt();

System.out.println("Enter internal marks for 5 subjects:");

int[] iMarks = new int[5];

for (int j = 0; j < 5; j++) {

iMarks[j] = sc.nextInt();

}

System.out.println("Enter external marks for 5 subjects:");

int[] eMarks = new int[5];

for (int j = 0; j < 5; j++) {

eMarks[j] = sc.nextInt();

}

sc.nextLine();

students[i] = new Externals(name, usn, sem, eMarks);

intMarks[i] = new Internals(iMarks);

}

System.out.println("\nStudent Details:");

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

students[i].displayDetails();

intMarks[i].displayIMarks();

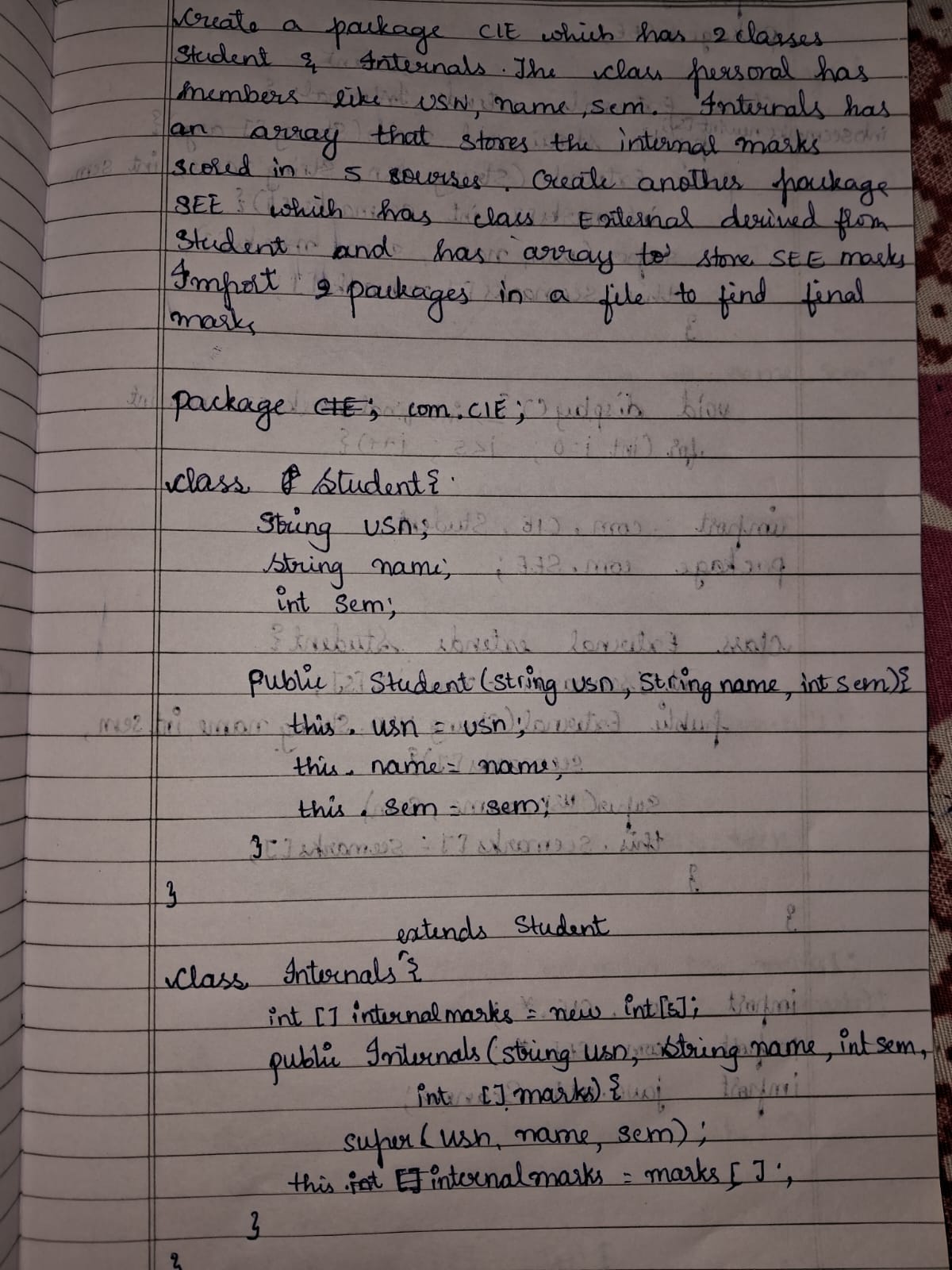
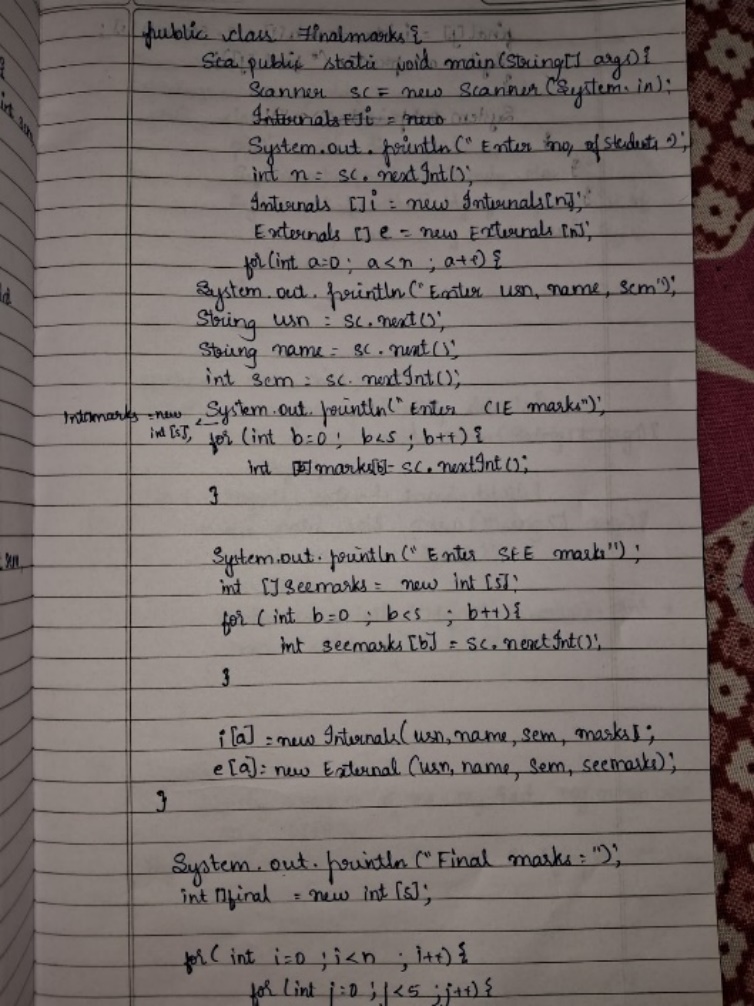
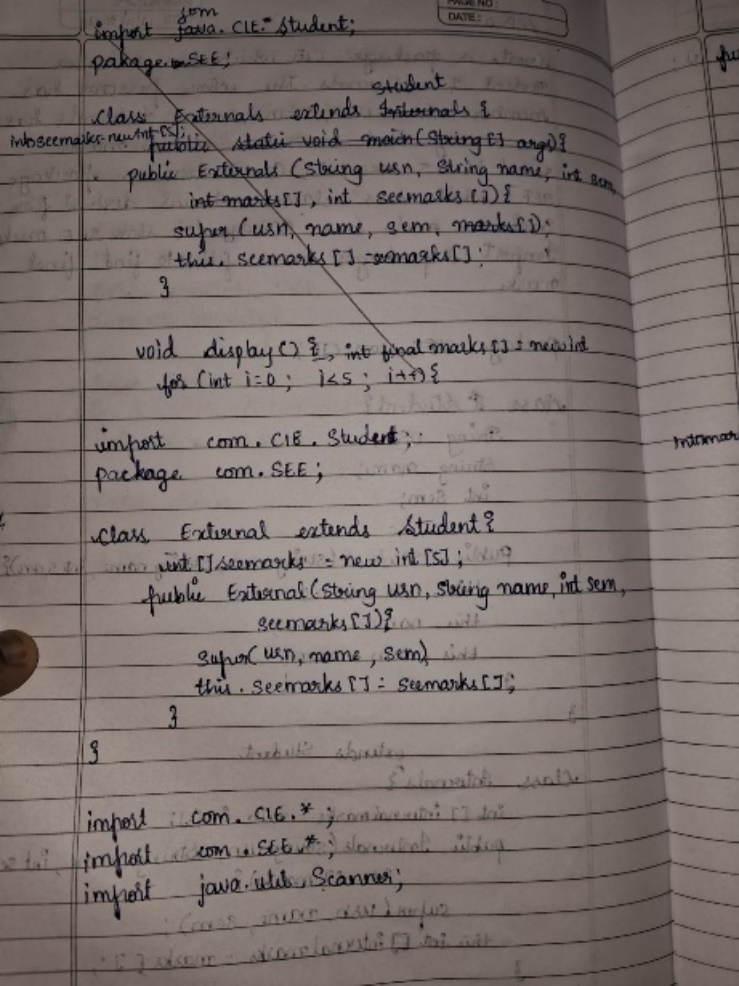
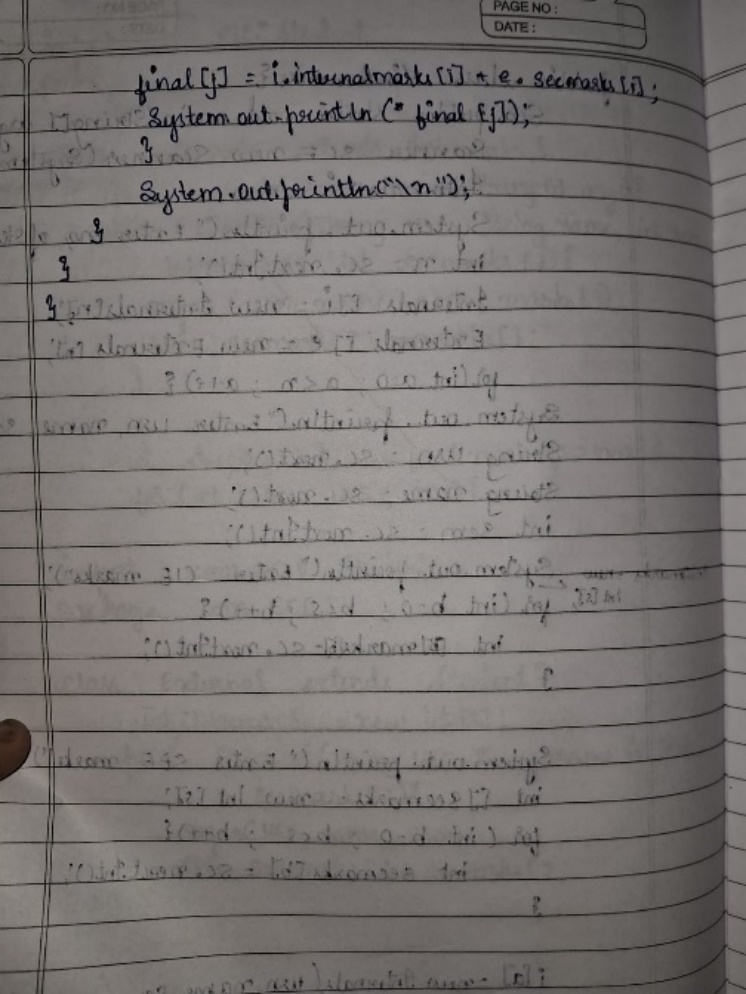
students[i].displayEMarks();

}

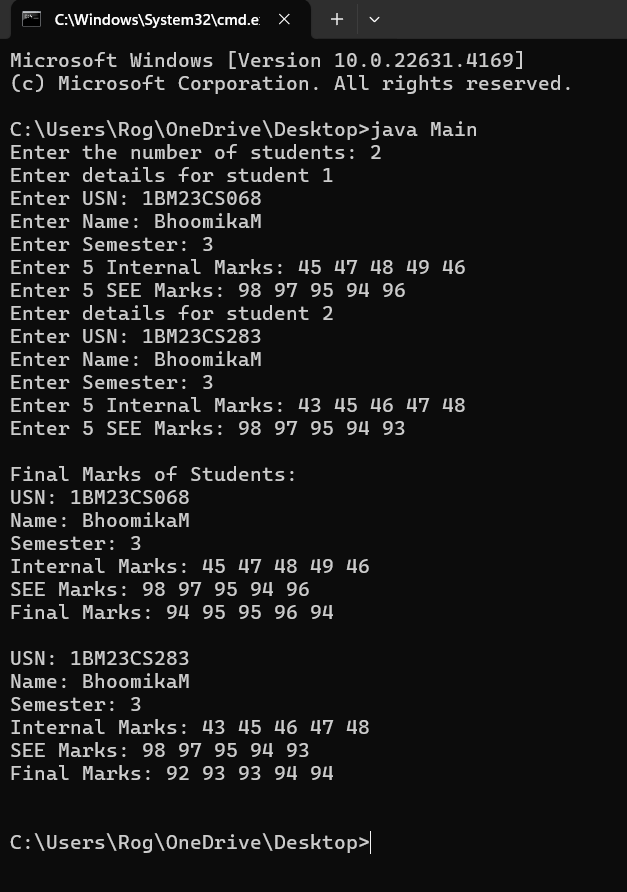
sc.close();

}

}

Output:



2.Family program:

Code:

package com.example.me;

public class Myself{

public void bhoomika(){

System.out.println(“I am Bhoomika BG\n My age is 19”);}

}

package com.example.family;

import com.example.me.Myself;

public class Family{

public static void main(String[] args){

Myself me=new Myself();

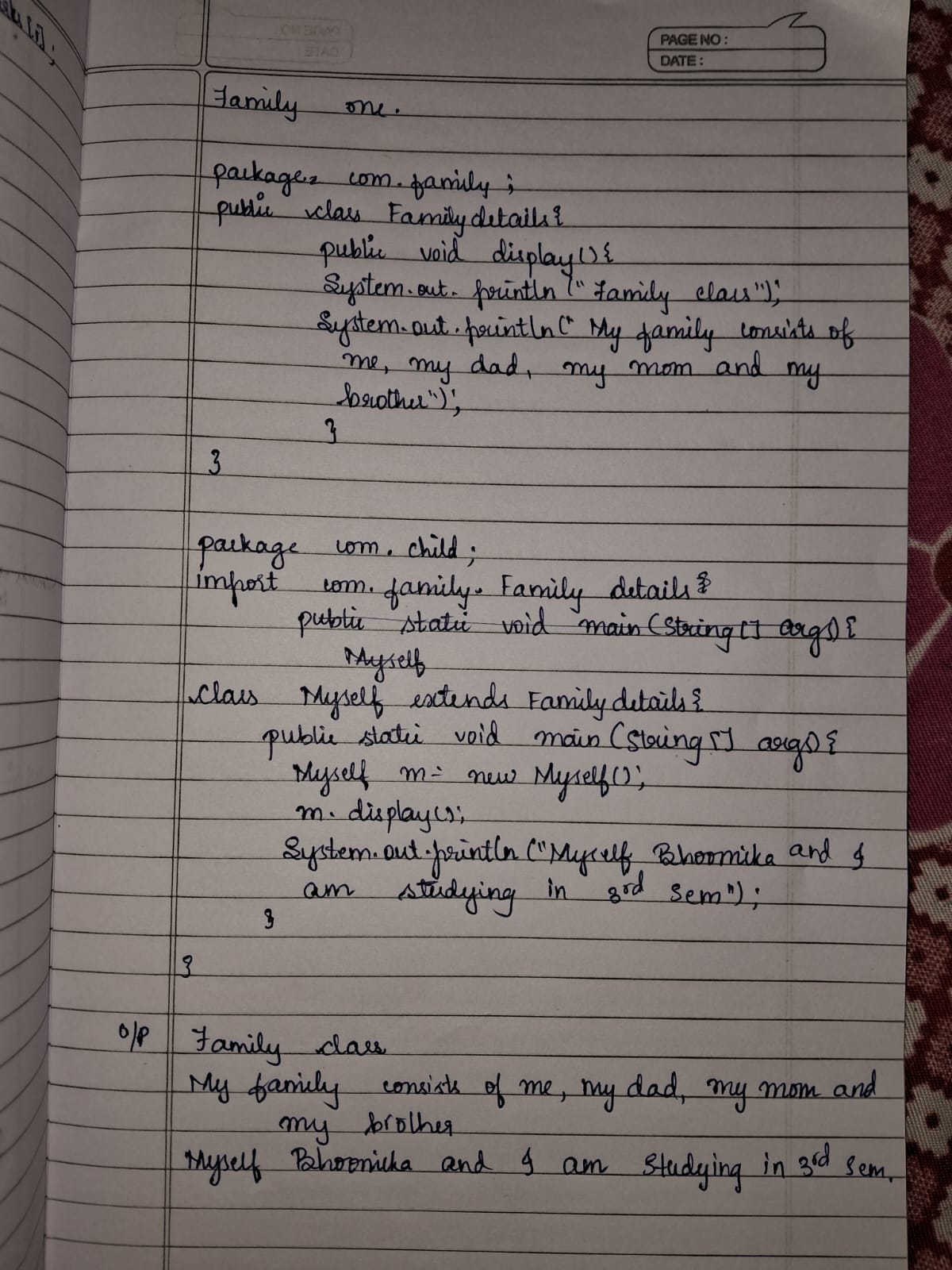
Family fam=new Family();

System.out.println(“MY Family has 3 members”;

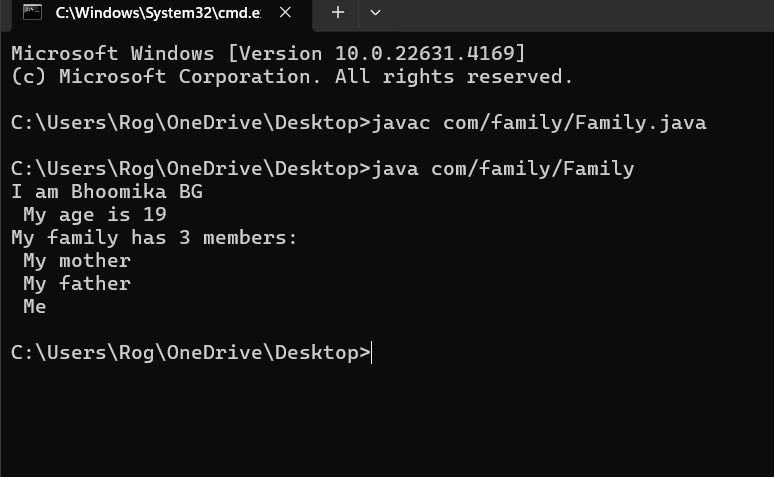
me.bhoomika();

}

}



Output:



Program 7

Interfaces

Code:

interface Polygon {

default double getPerimeter(sides) {

double perimeter = 0;

for (double side : sides) {

perimeter += side;

}

return perimeter;

}

double getArea();

}

class Rectangle implements Polygon {

private double length;

private double breadth;

Rectangle(double length, double breadth) {

this.length = length;

this.breadth = breadth;

}

public double getArea() {

return length \* breadth;

}

public double calculatePerimeter() {

return getPerimeter(length, breadth, length, breadth);

}

}

class Triangle implements Polygon {

private double base;

private double height;

private double side1, side2, side3;

Triangle(double base, double height, double side1, double side2, double side3) {

this.base = base;

this.height = height;

this.side1 = side1;

this.side2 = side2;

this.side3 = side3;

}

public double getArea() {

return 0.5 \* base \* height;

}

public double calculatePerimeter() {

return getPerimeter(side1, side2, side3);

}

}

// Main Class

public class Main {

public static void main(String[] args) {

// Rectangle instance

Rectangle rectangle = new Rectangle(5.0, 3.0);

System.out.println("Rectangle:");

System.out.println("Area: " + rectangle.getArea());

System.out.println("Perimeter: " + rectangle.calculatePerimeter());

// Triangle instance

Triangle triangle = new Triangle(5.0, 4.0, 3.0, 4.0, 5.0);

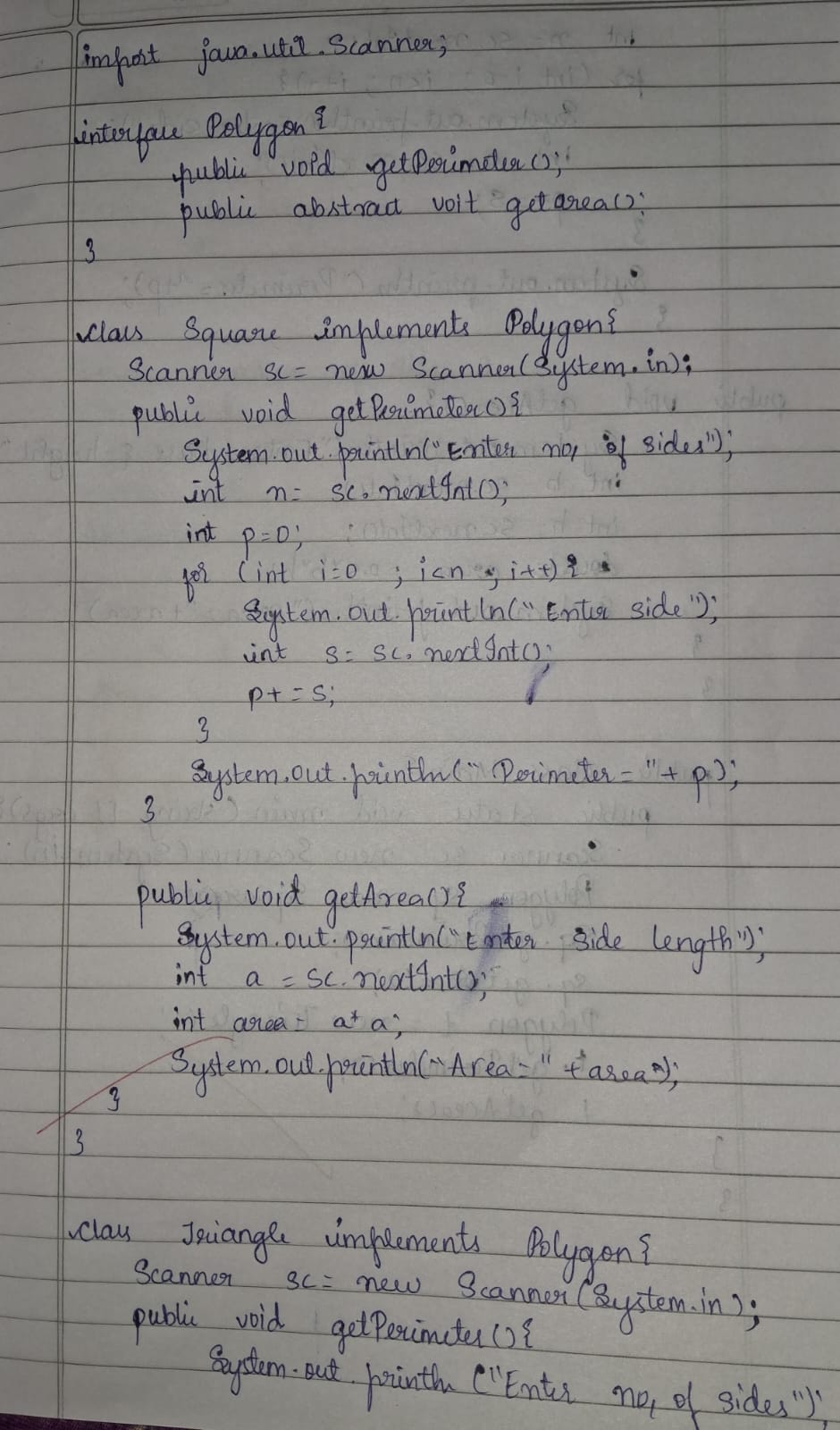
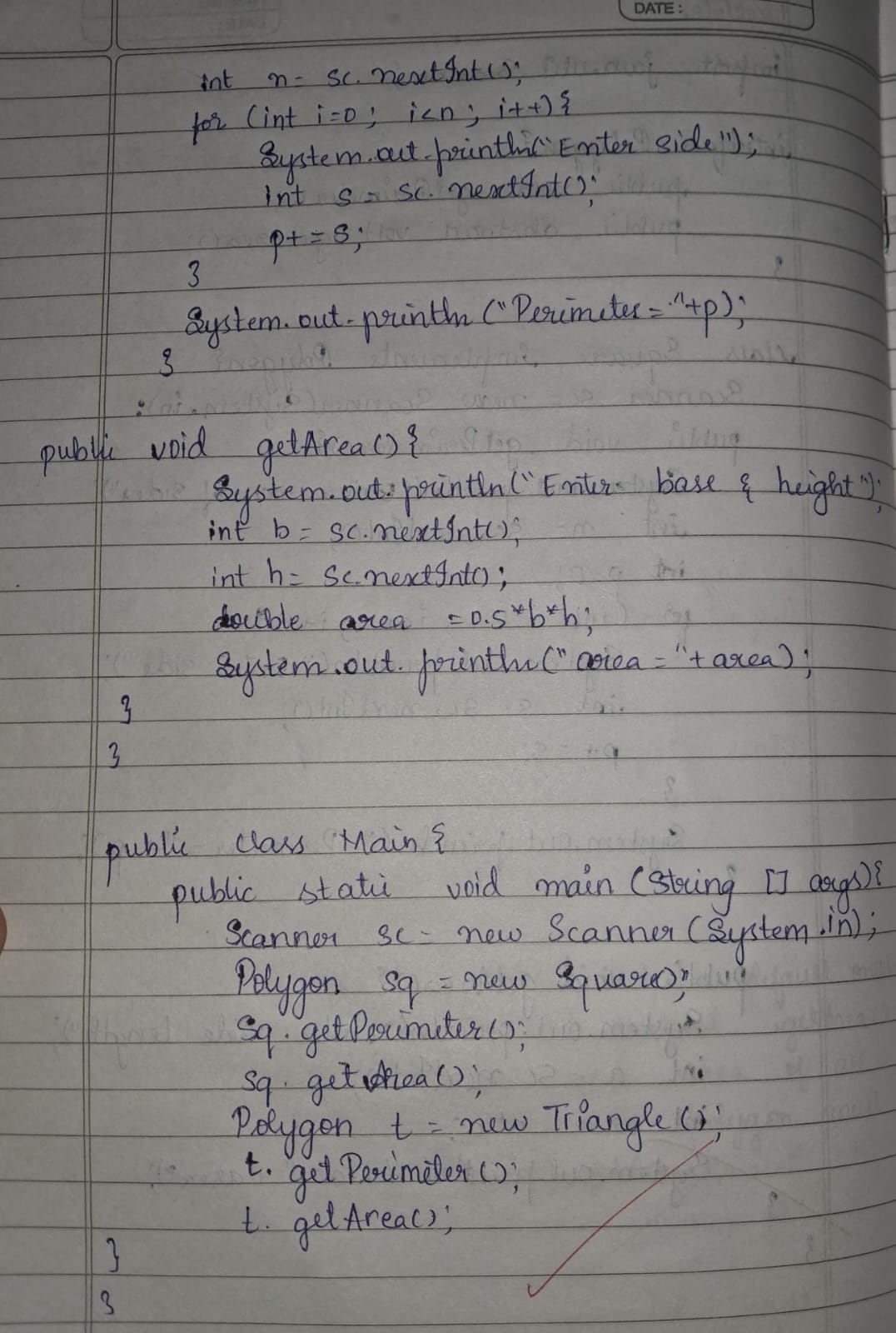
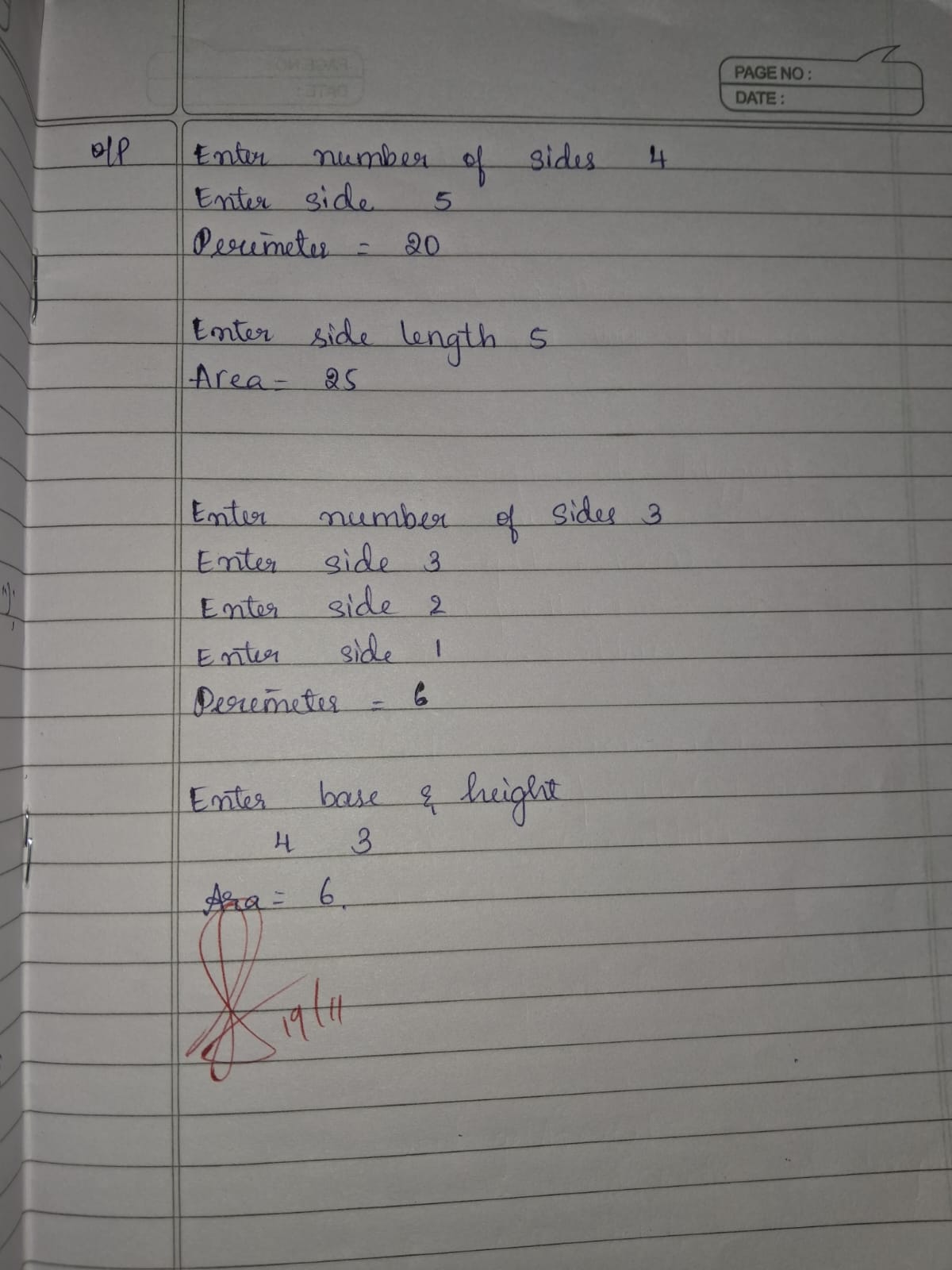
System.out.println("\nTriangle:");

System.out.println("Area: " + triangle.getArea());

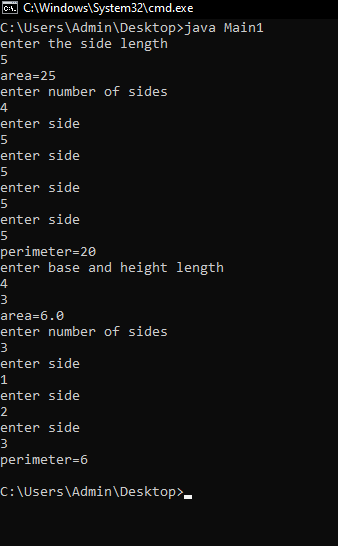
System.out.println("Perimeter: " + triangle.calculatePerimeter());

    }

}

Output:



Program 8

Exception Handling

Code:

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 less than zero.");

}

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 less than zero.");

}

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 ExceptionHandlingInheritance {

public static void main(String[] args) {

try {

Son son1 = new Son(40, 15);

son1.displayAges();

Father fatherInvalid = new Father(-5);

Son sonInvalid = new Son(35, 40);

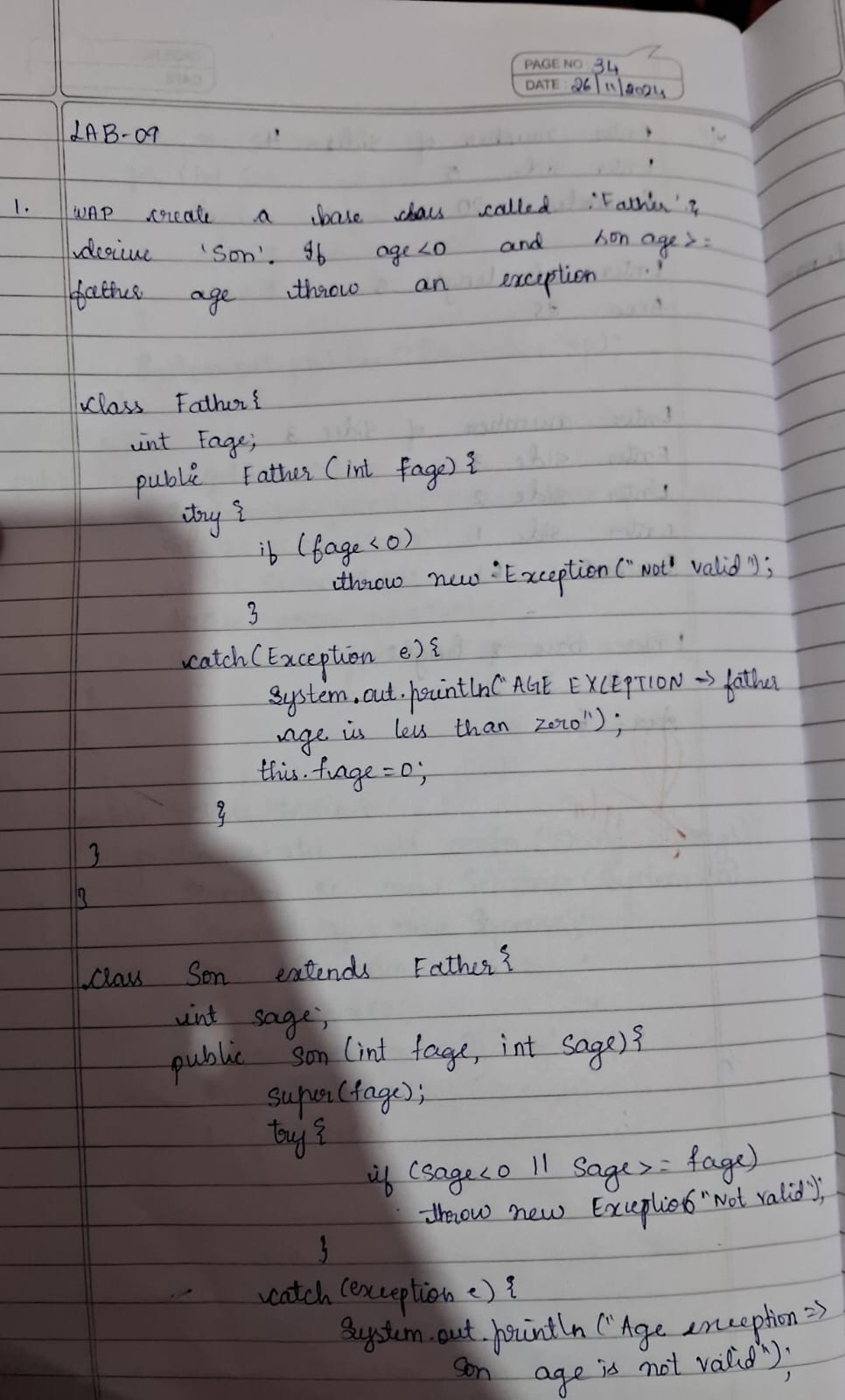
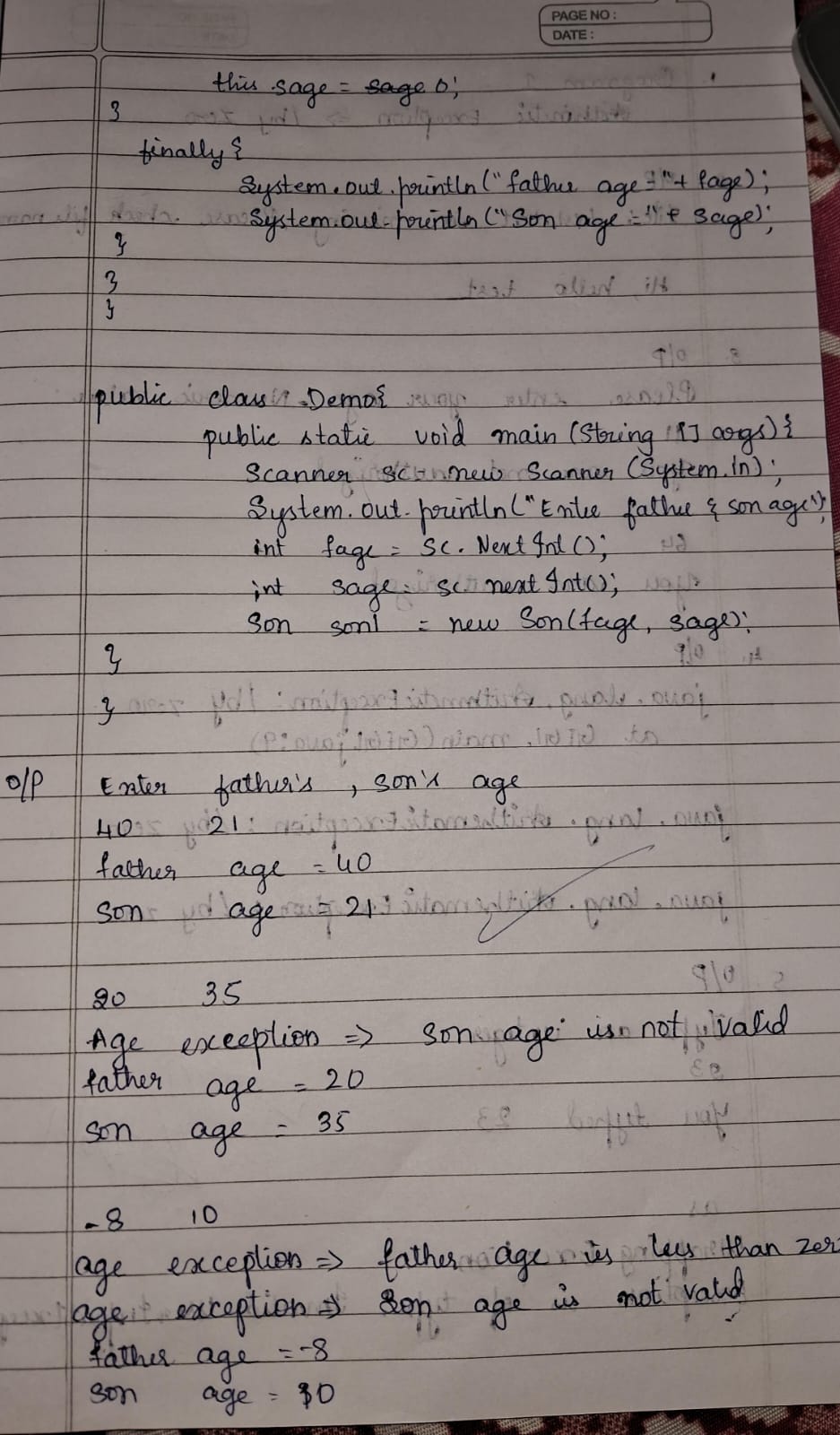
} catch (WrongAgeException e) {

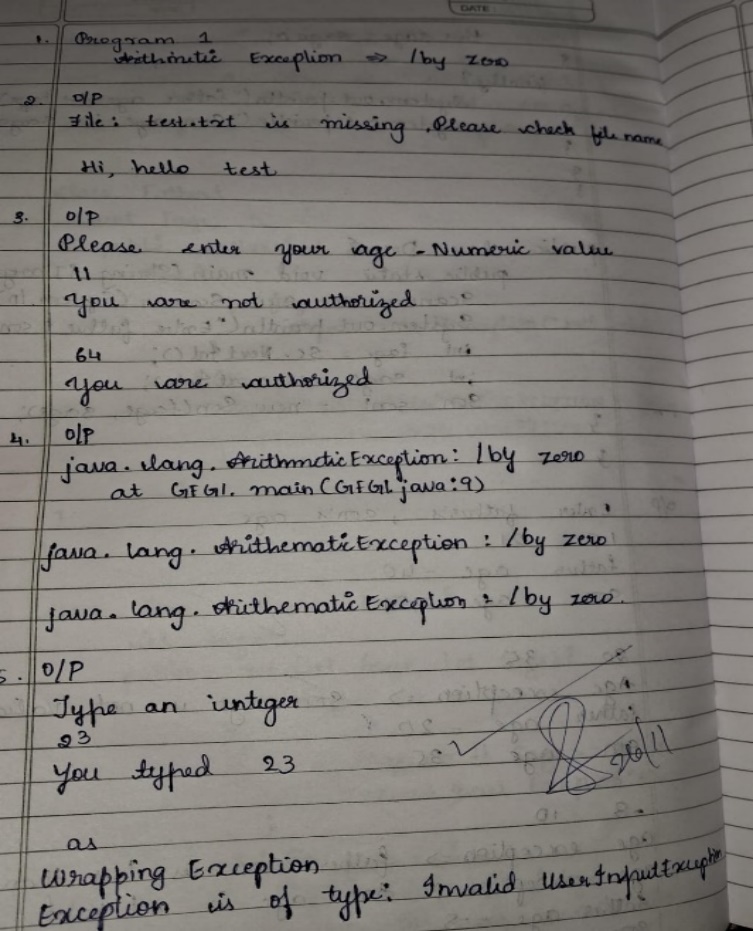
System.out.println("Exception occurred: " + e.getMessage());

       }

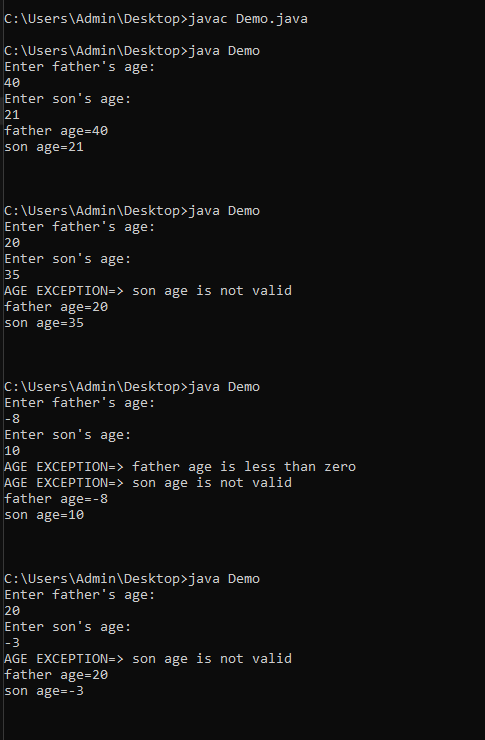
    }

}



Output:



Program 9

Thread Programming

Code:

class Thread1 extends Thread {

public void run() {

try {

while (true) {

System.out.println("BMS College of Engineering");

Thread.sleep(10000);

}

} catch (InterruptedException e) {

System.out.println("Thread1 interrupted: " + e.getMessage());

}

}

}

class Thread2 extends Thread {

public void run() {

try {

while (true) {

System.out.println("CSE");

Thread.sleep(2000);

}

} catch (InterruptedException e) {

System.out.println("Thread2 interrupted: " + e.getMessage());

}

}

}

public class MultiThreadDisplay {

public static void main(String[] args) {

Thread1 t1 = new Thread1();

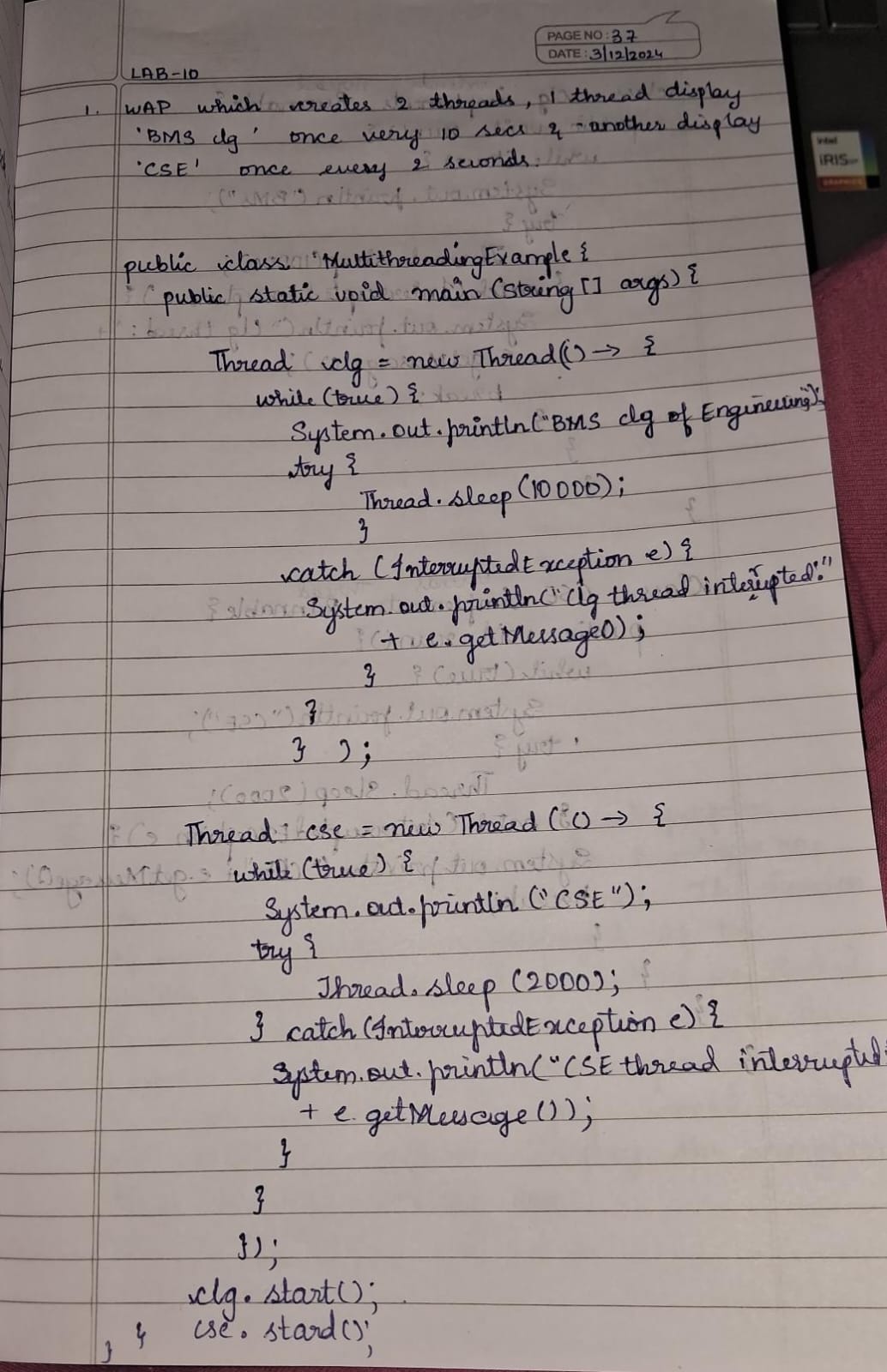
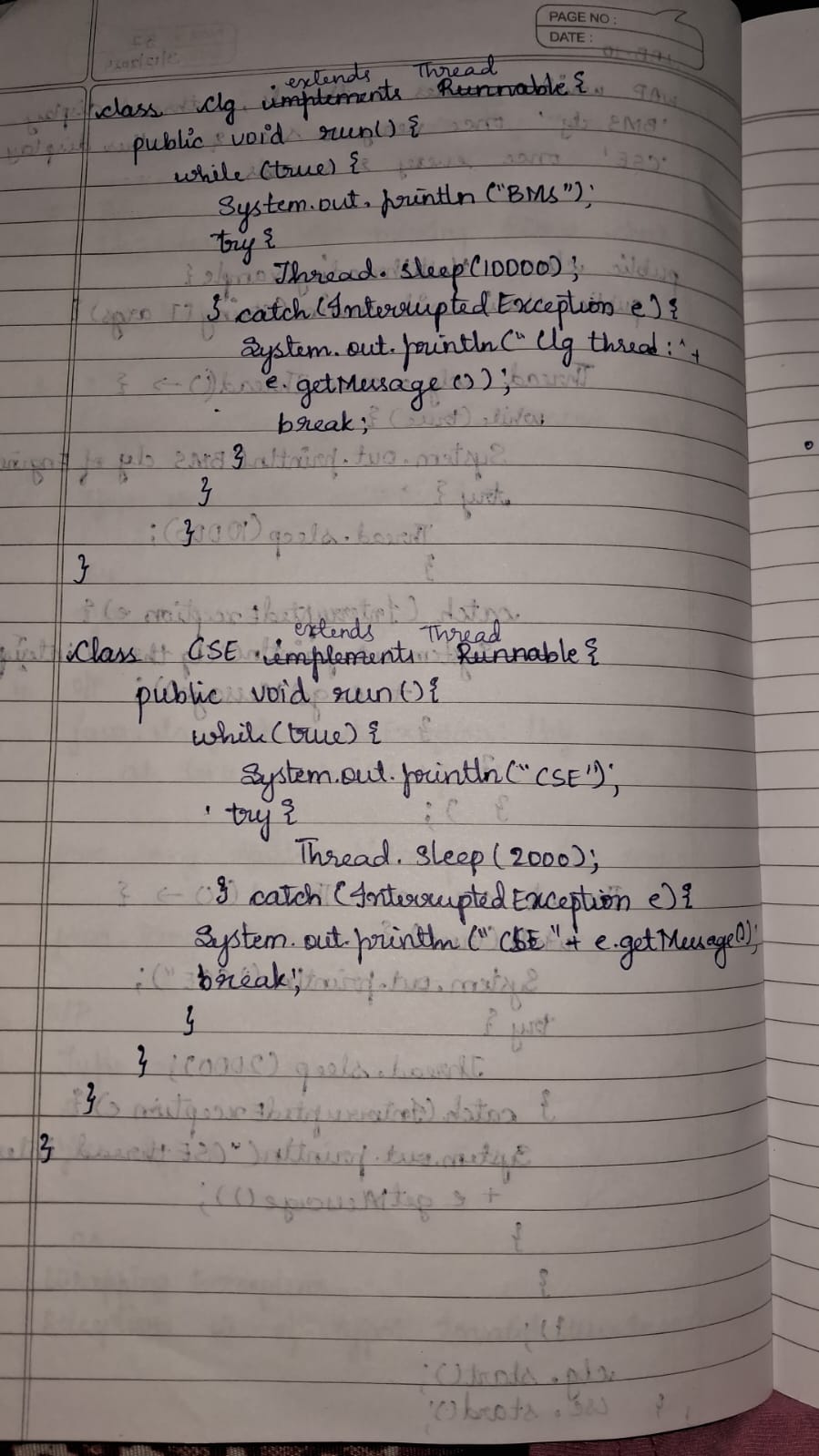
t1.start();

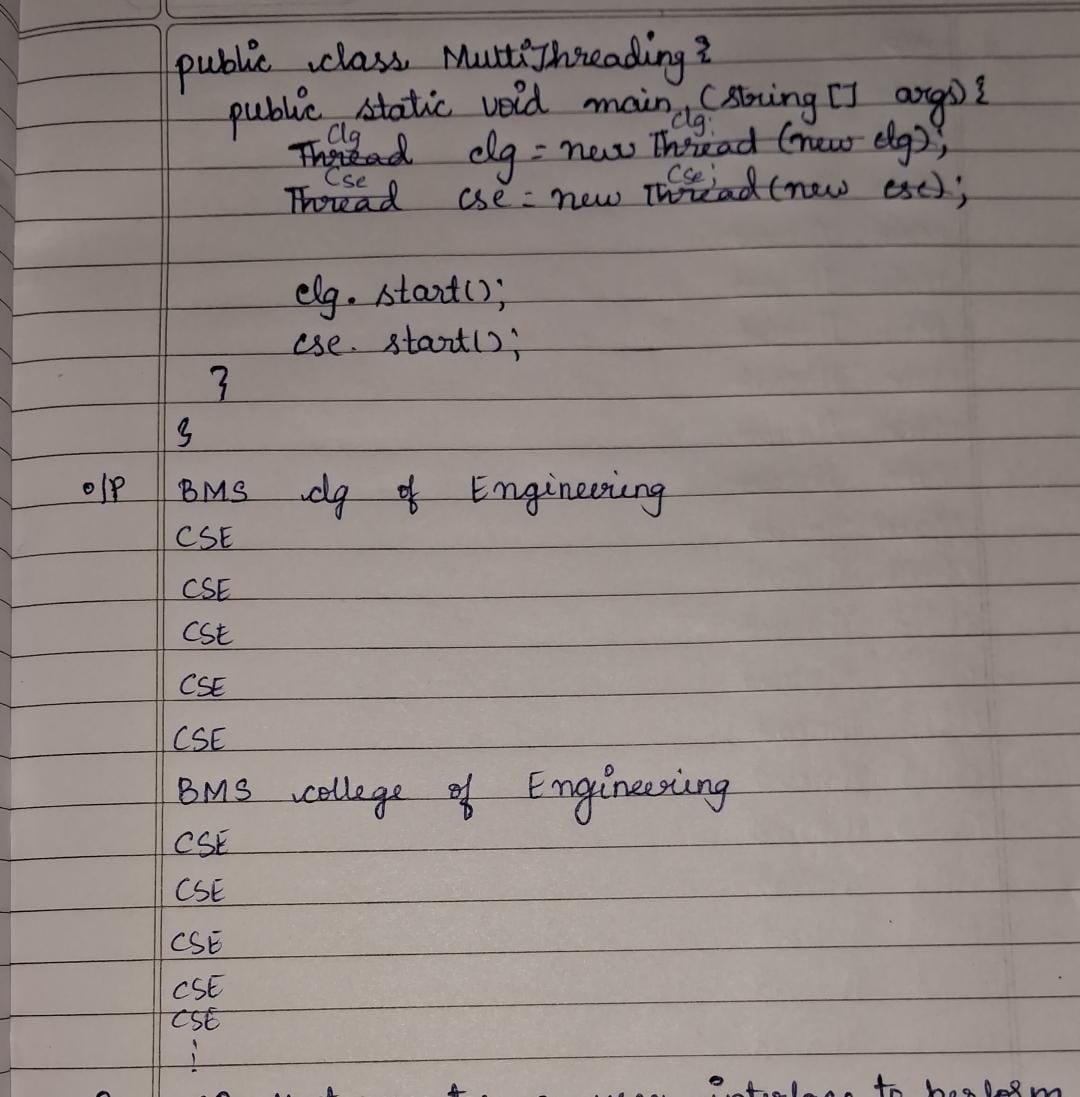
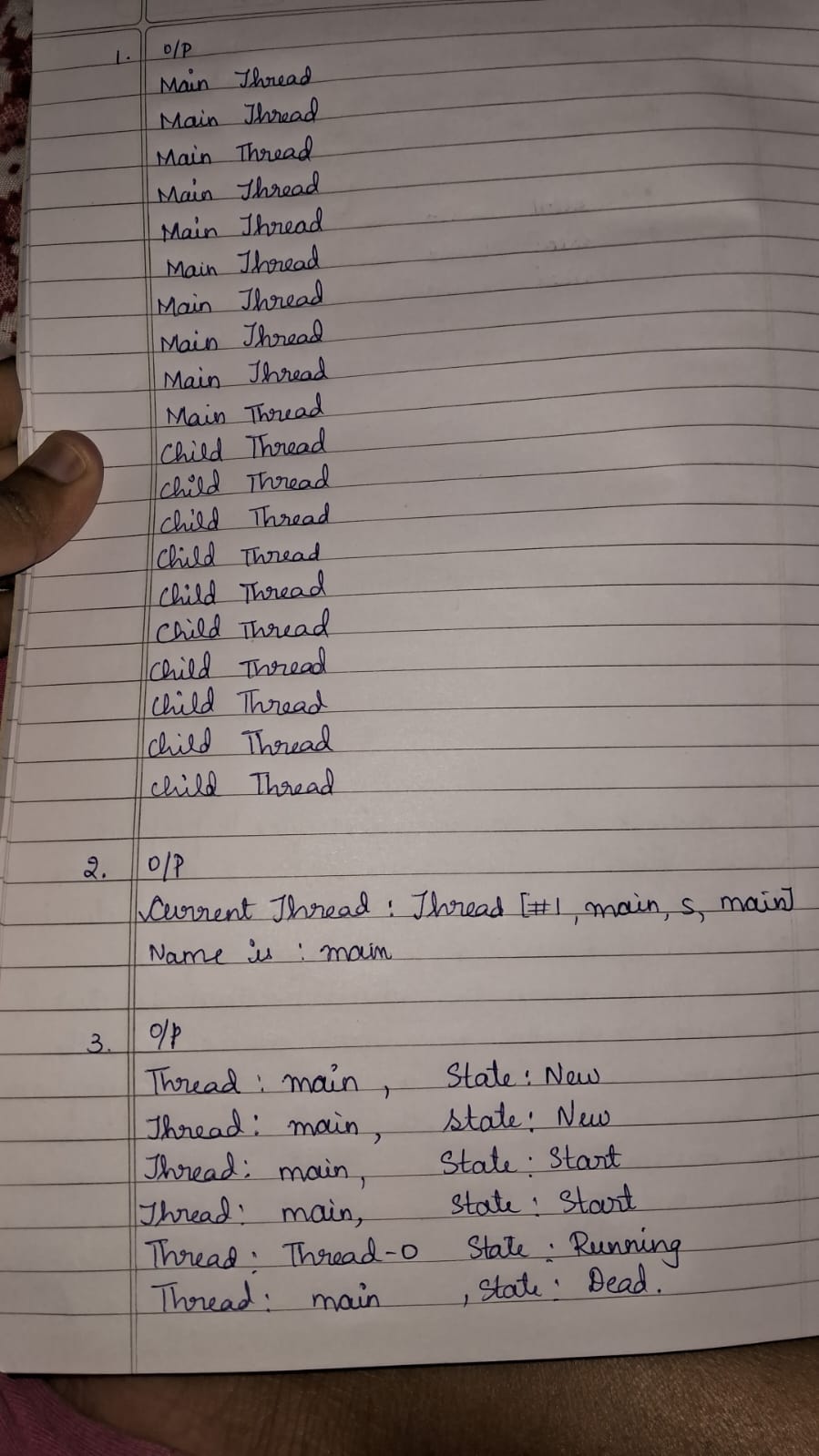
Thread2 t2 = new Thread2();

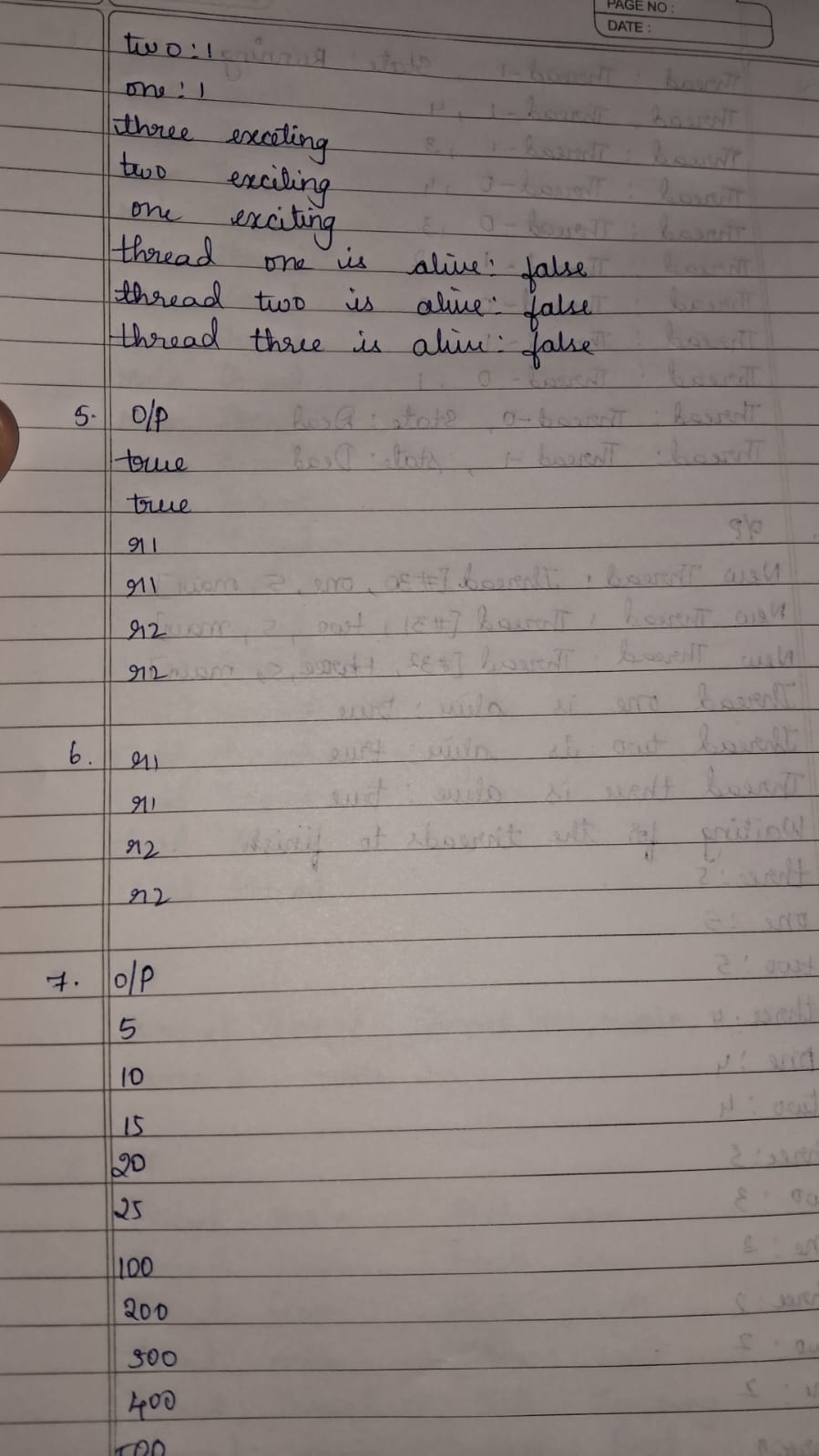
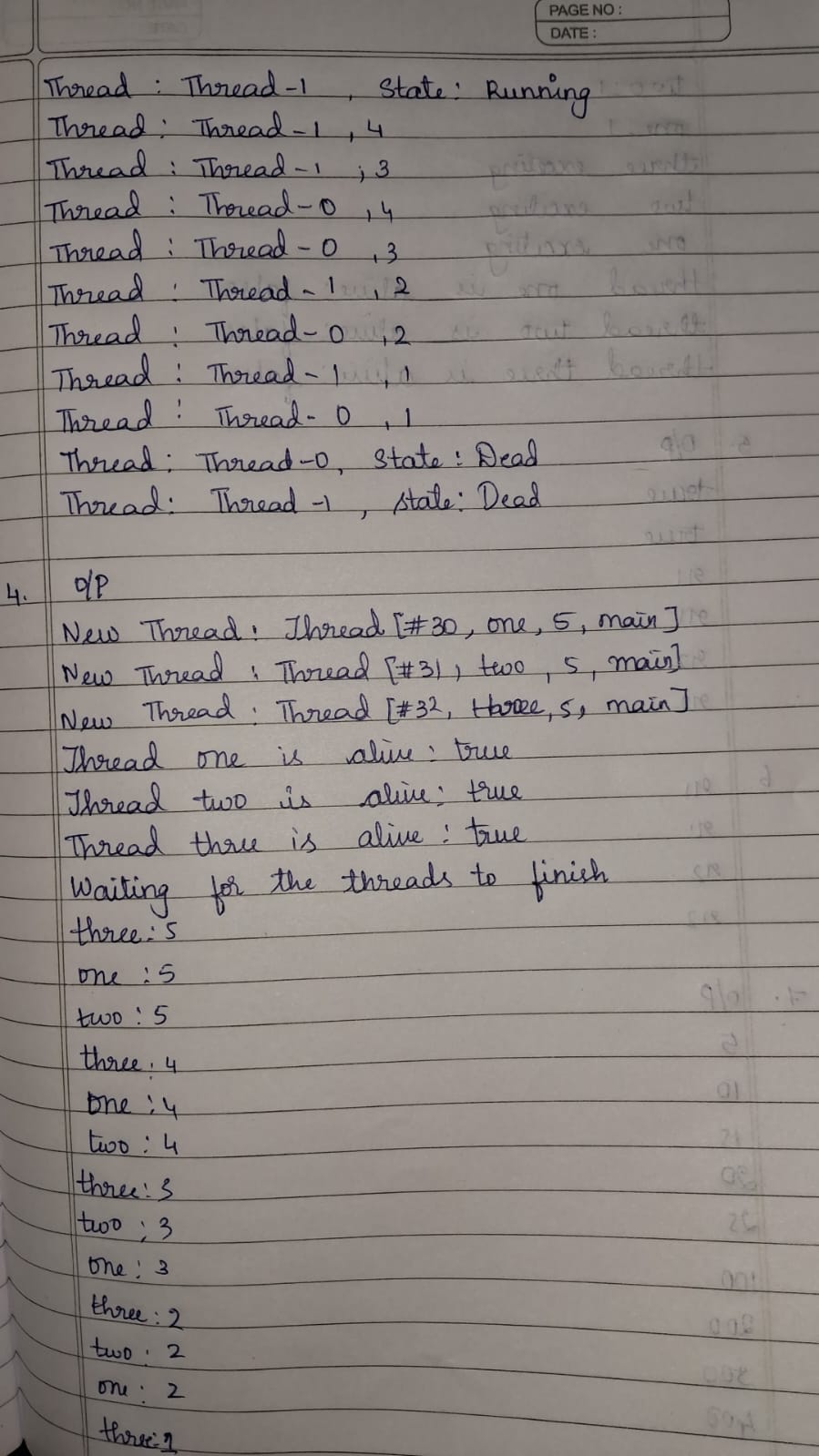
t2.start();

    }

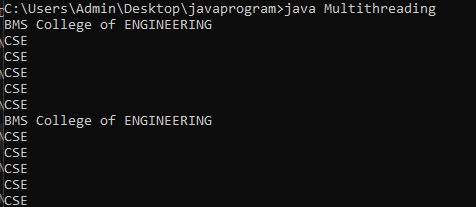
}

** **

** **

** **

Output:



Program 10

Open Ended Exercise

Code:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class DivisionApp {

public static void main(String[] args)

JFrame frame = new JFrame("Division Calculator");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(400, 200);

frame.setLayout(new GridLayout(4, 2, 10, 10));

JLabel labelNum1 = new JLabel("Enter number 1:");

JTextField textNum1 = new JTextField();

JLabel labelNum2 = new JLabel("Enter number 2:");

JTextField textNum2 = new JTextField();

JLabel labelResult = new JLabel("Result:");

JTextField textResult = new JTextField();

textResult.setEditable(false);

JButton buttonDivide = new JButton("Divide");

frame.add(labelNum1);

frame.add(textNum1);

frame.add(labelNum2);

frame.add(textNum2);

frame.add(labelResult);

frame.add(textResult);

frame.add(new JLabel());

frame.add(buttonDivide);

buttonDivide.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

int num1 = Integer.parseInt(textNum1.getText());

int num2 = Integer.parseInt(textNum2.getText());

if (num2 == 0) {

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

}

int result = num1 / num2;

textResult.setText(String.valueOf(result));

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(frame,

"Invalid input! Please enter integers only.",

"Number Format Error",

JOptionPane.ERROR\_MESSAGE);

} catch (ArithmeticException ex) {

JOptionPane.showMessageDialog(frame,

ex.getMessage(),

"Arithmetic Error",

JOptionPane.ERROR\_MESSAGE);

}

}

});

frame.setVisible(true);

    }

}

Output:

