VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Aman Vats (1BM23CS026)

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 **Aman Vats (1BM23CS026),** 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.

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

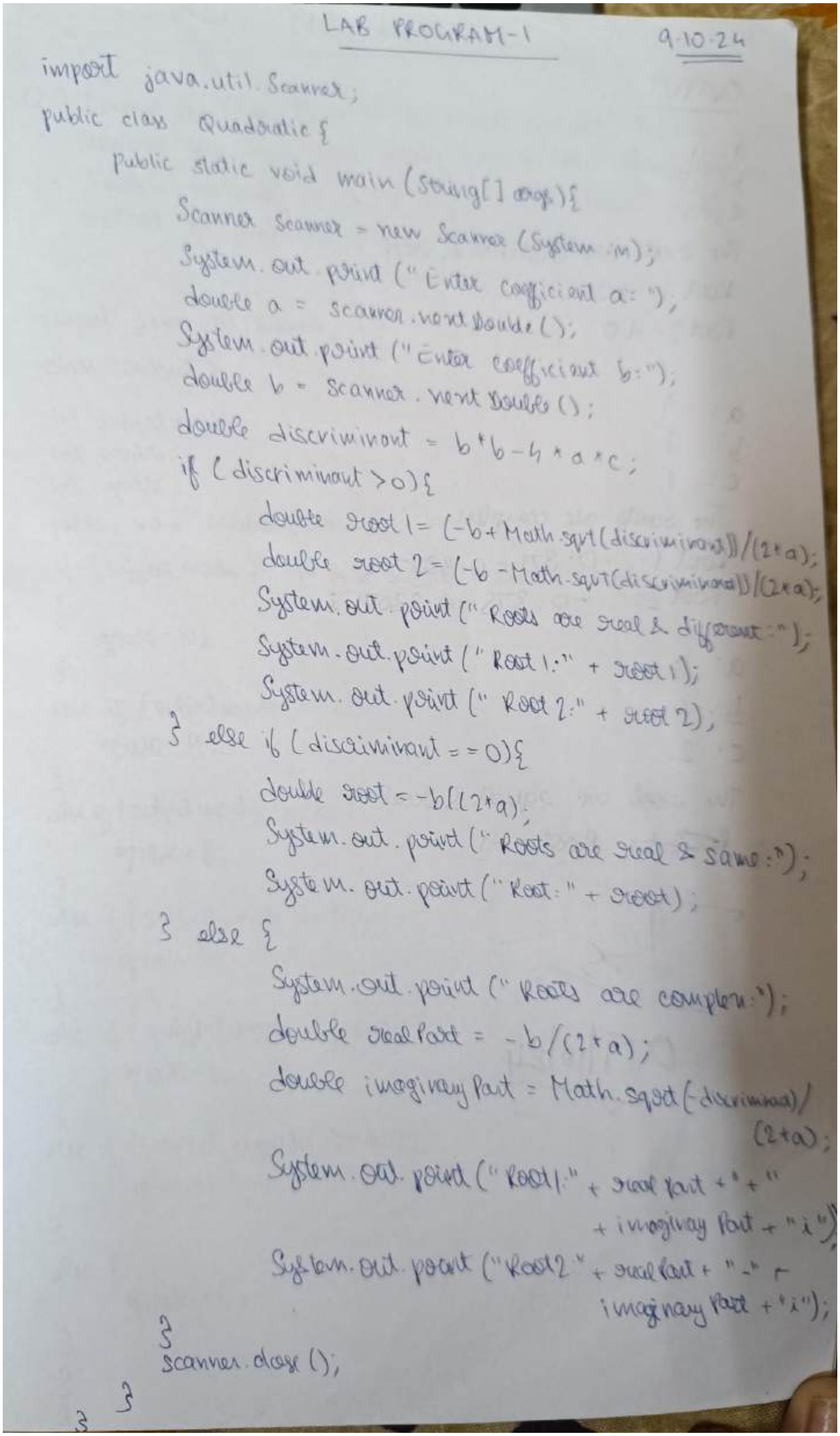
Index

Github Link:
https://github.com/amanvats7/Java-Lab/tree/main

Program 1

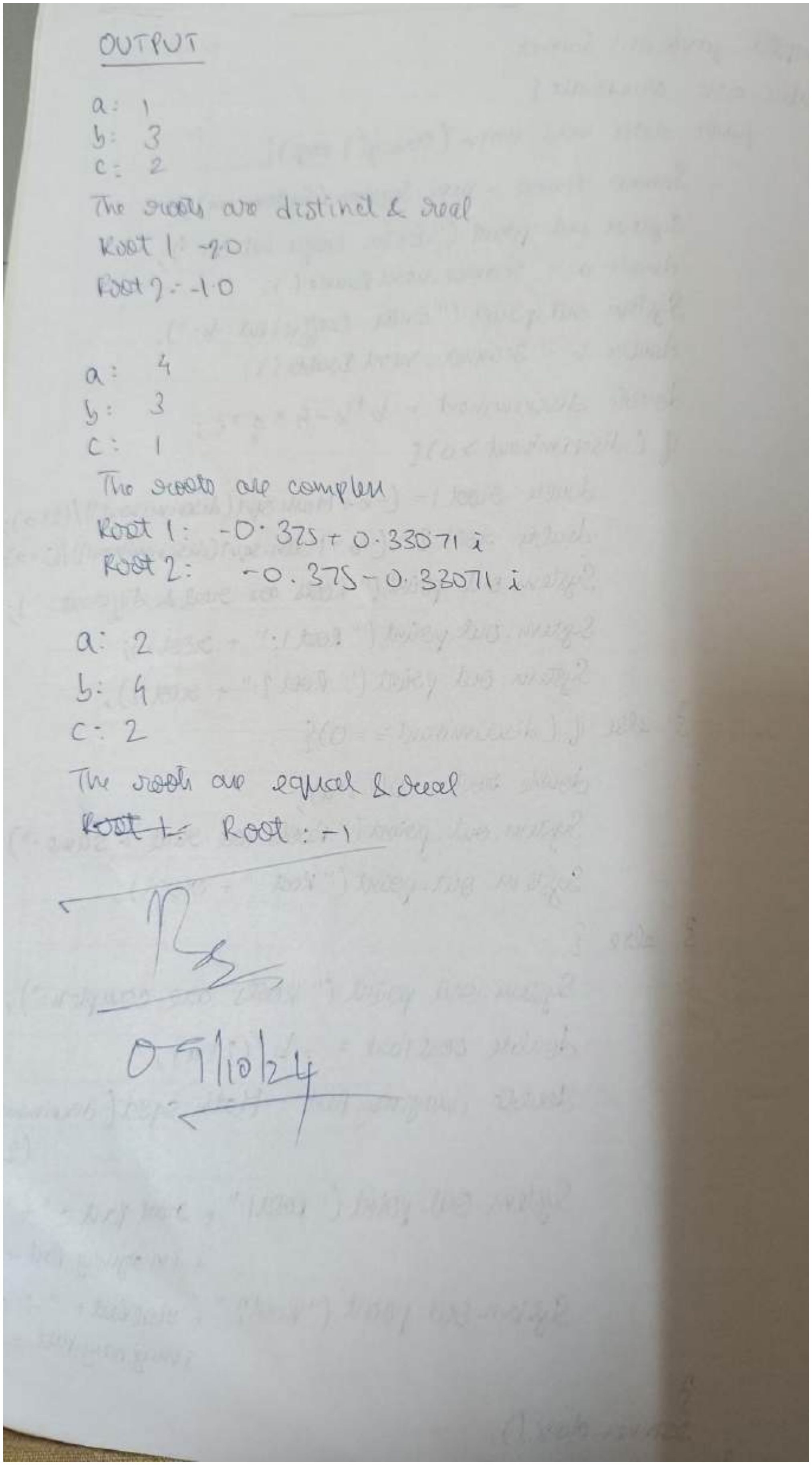
Implement Quadratic Equation

Observation:



LAB PROGRAM-1 9-10-24

```
import java.util.Scanner;
public class Quadratic {
    public static void main(String[] args){
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter coefficient a: ");
        double a = scanner.nextDouble();
        System.out.print("Enter coefficient b: ");
        double b = scanner.nextDouble();
        double discriminant = b*b-4*a*c;
        if (discriminant >0){
            double root1= (-b+Math.sqrt(discriminant))/(2*a);
            double root2= (-b -Math.sqrt(discriminant))/(2*a);
            System.out.print("Roots are real & different:");
            System.out.print("Root1:" + root1);
            System.out.print(" Root2:" + root2),
        } else if (discriminant = = 0){
            double root = -b((2*a);
            System.out.print("Roots are real & same:");
            System.out.print("Root:" + root);
        } else {
            System.out.print("Roots are complex:");
            double realPart = -b/(2*a);
            double imaginaryPart = Math.sqrt(-discriminant)/
                                                        (2*a);
            System.out.print("Root1:" + realPart + " + "
                                  + imaginaryPart + "i")
            System.out.print("Root2:" + realPart + " - " +
                                  imaginaryPart + "i");
        }
        scanner.close();
    }
}
```



OUTPUT

```
a: 1
b: 3
c: 2
The roots are distinct & real
Root 1: -2.0
Root 2: -1.0

a: 4
b: 3
c: 1
The roots are complex
Root 1: -0.375 + 0.33071 i
Root 2:  -0.375 - 0.33071 i

a: 2
b: 4
c: 2
The roots are equal & real
```

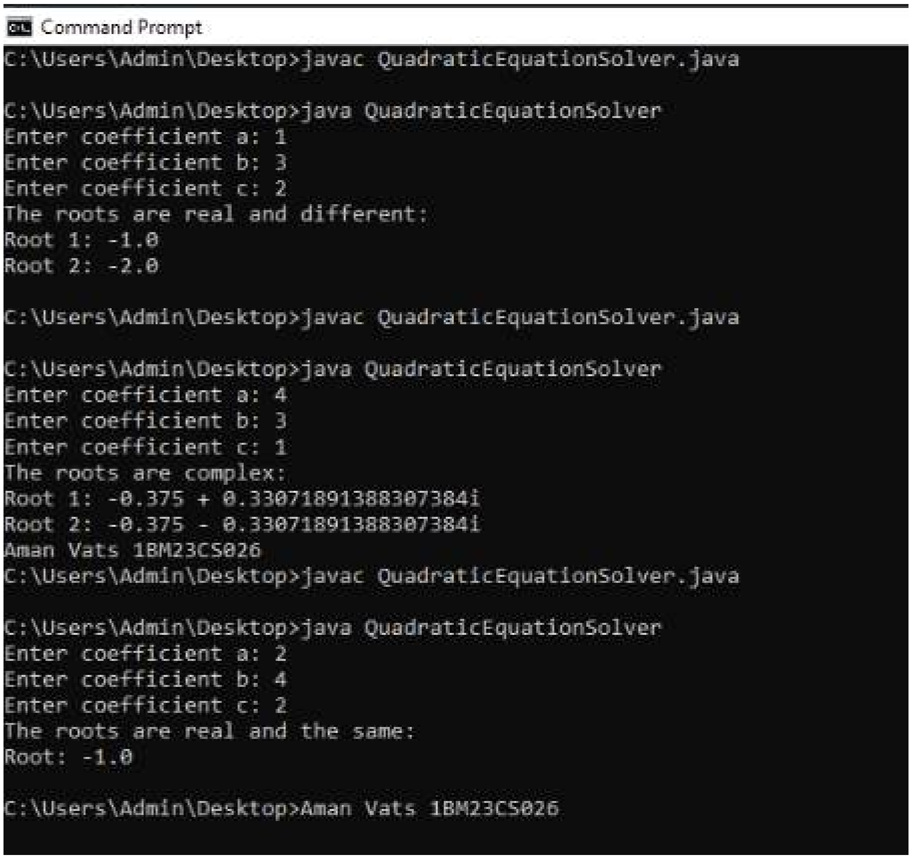
~~Root 1:~~ Root : -1

09/10/24

Code:

```
import java.util.Scanner;

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

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

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

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

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

        if (discriminant > 0) {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("The roots are real and different:");
            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 the same:");
            System.out.println("Root: " + root);
        } else {
            System.out.println("The roots are complex:");
            double realPart = -b / (2 * a);
            double imaginaryPart = Math.sqrt(-discriminant) / (2 * a);
            System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
            System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");
                    System.out.print("Aman Vats 1BM23CS026");
        }

        scanner.close();
    }
}
```

Output:

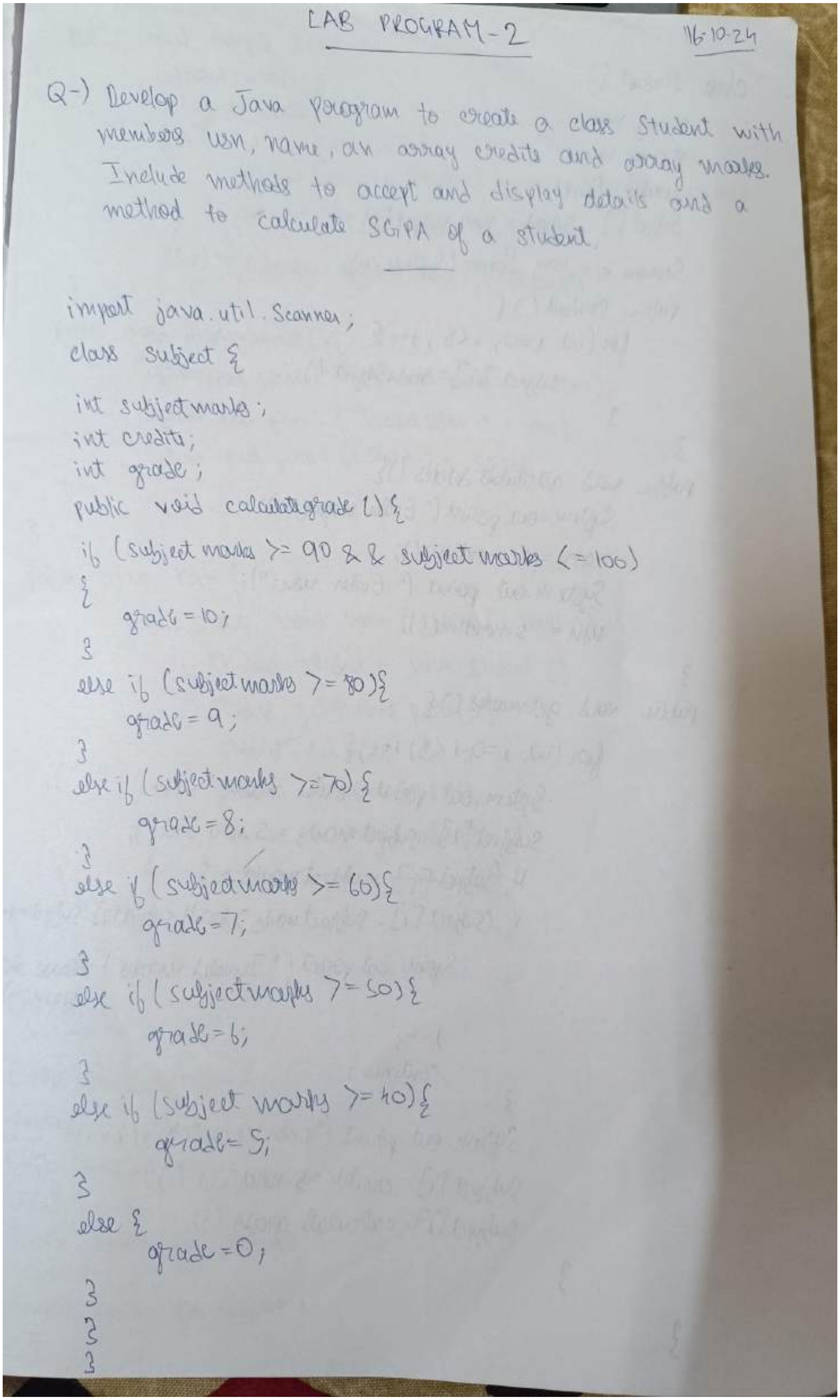


```
C:\Users\Admin\Desktop>javac QuadraticEquationSolver.java

C:\Users\Admin\Desktop>java QuadraticEquationSolver
Enter coefficient a: 1
Enter coefficient b: 3
Enter coefficient c: 2
The roots are real and different:
Root 1: -1.0
Root 2: -2.0

C:\Users\Admin\Desktop>javac QuadraticEquationSolver.java

C:\Users\Admin\Desktop>java QuadraticEquationSolver
Enter coefficient a: 4
Enter coefficient b: 3
Enter coefficient c: 1
The roots are complex:
Root 1: -0.375 + 0.33071891388307384i
Root 2: -0.375 - 0.33071891388307384i
Aman Vats 1BM23CS026
C:\Users\Admin\Desktop>javac QuadraticEquationSolver.java

C:\Users\Admin\Desktop>java QuadraticEquationSolver
Enter coefficient a: 2
Enter coefficient b: 4
Enter coefficient c: 2
The roots are real and the same:
Root: -1.0

C:\Users\Admin\Desktop>Aman Vats 1BM23CS026
```

Program 2

SGPA CALCULATOR

Observation:

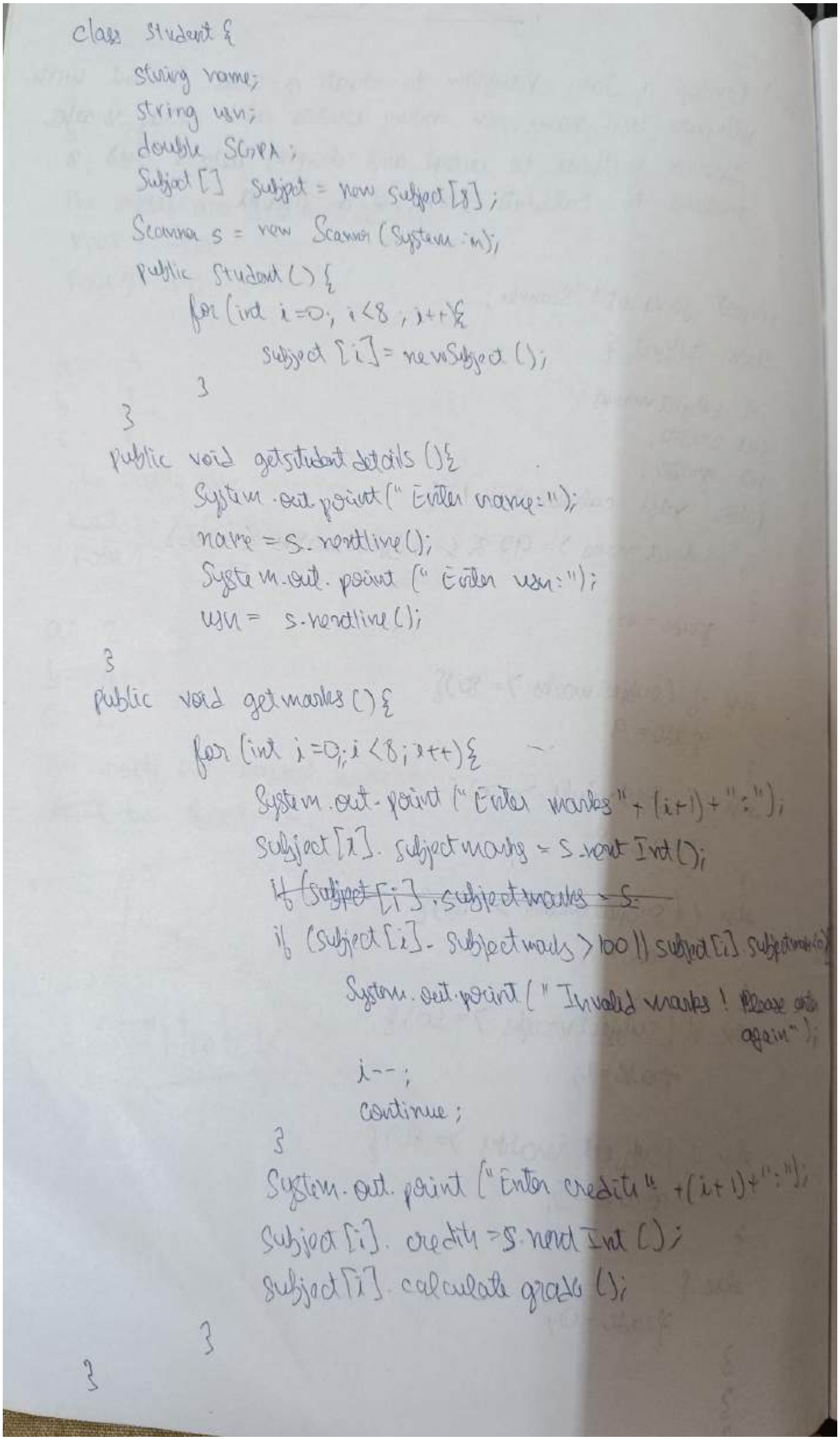


LAB PROGRAM-2

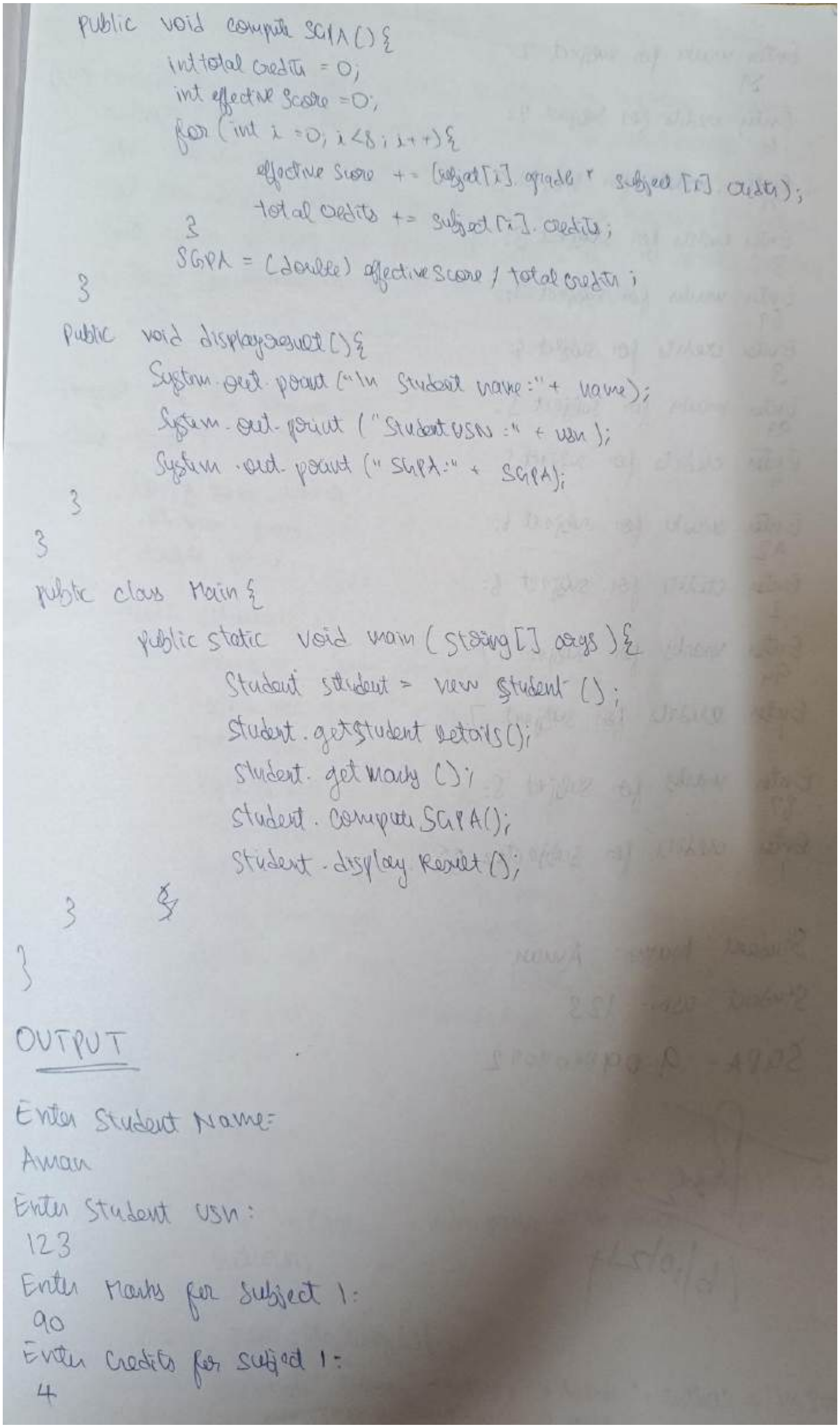
16-10-24

Q-) Develop a Java program to create a class Student with members usn, name, an array credits and array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;
class subject {
int subjectmarks;
int credits;
int grade;
public void calculategrade () {
 if (subject marks >= 90 && subject marks <= 100)
 {
    grade = 10;
 }
 else if (subjectmarks >= 80){
    grade = 9;
 }
 else if (subjectmarks >=70) {
     grade=8;
 }
 else if (subjectmarks >= 60){
     grade=7;
 }
 else if (subjectmarks >= 50) {
      grade=6;
 }
 else if (subject marks >= 40){
      grade= 5;
 }
 else {
      grade =0;
 }
 }
 }
```



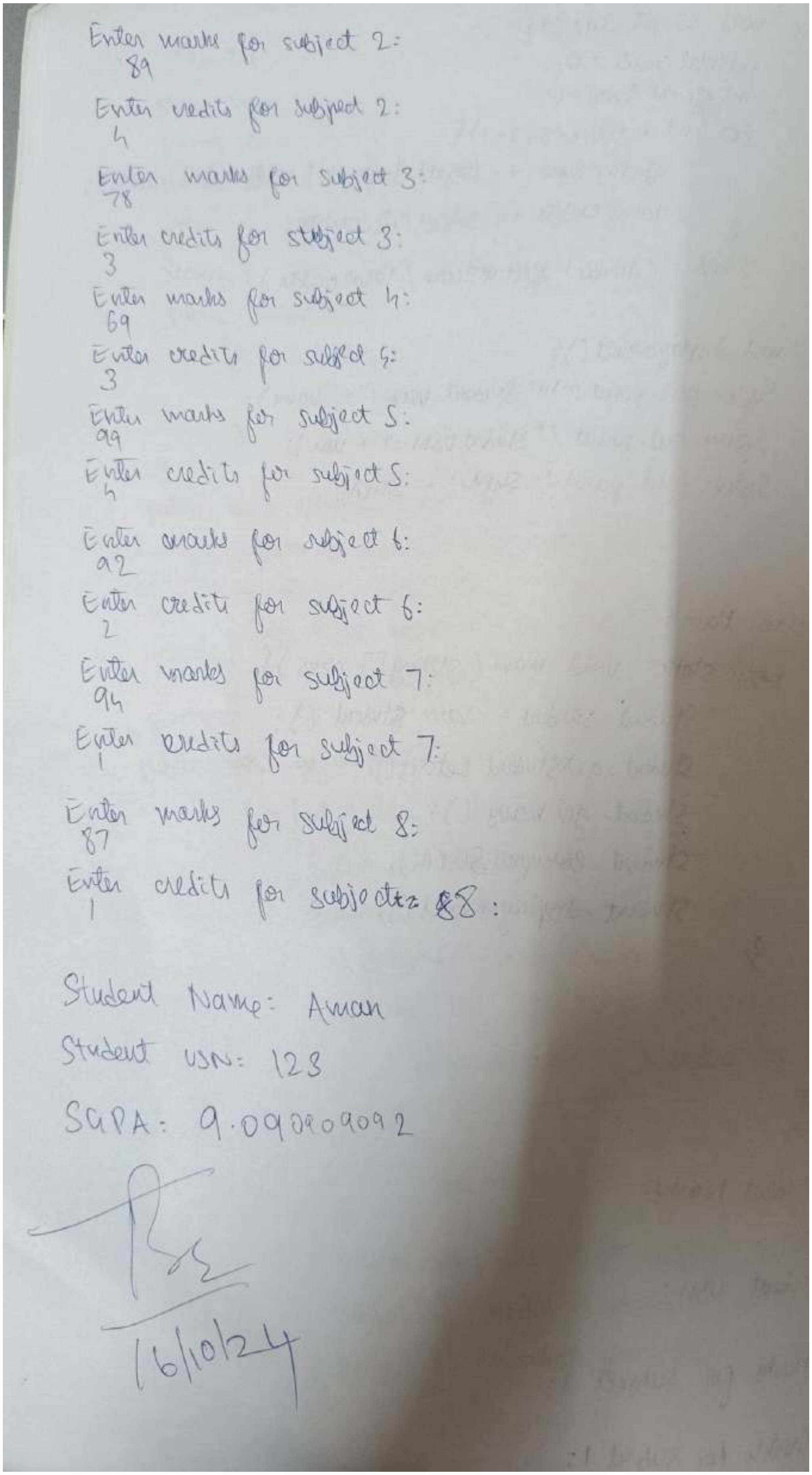
```
class Student {
    String name;
    String usn;
    double SGPA;
    Subject[] subject = new Subject[8];
    Scanner s = new Scanner(System.in);
    public Student() {
        for (int i=0; i<8; i++){
            subject[i] = newSubject();
        }
    }
  public void getstudentdetails(){
        System.out.print("Enter name:");
        name = s.nextline();
        System.out.print("Enter usn:");
        usn = s.nextline();
  }
 public void getmarks(){
        for (int i=0; i<8; i++){
            System.out.print("Enter marks"+(i+1)+":");
            subject[i].subjectmarks = s.nextInt();
            ~~if (subject[i].subjectmarks = s.~~
            if (subject[i].subjectmarks>100|| subject[i].subjectmarks<0)
                System.out.print("Invalid marks! Please enter
                                                    again");

                i--;
                continue;
            }
            System.out.print("Enter credits"+(i+1)+":");
            subject[i].credits = s.nextInt();
            subject[i].calculategrade();

        }
}
```



```
    public void compute SGPA(){
            int total credits = 0;
            int effective Score = 0;
            for (int i = 0; i<8; i++){
                    effective Score += (subject[i].grade * subject[i].credits);
                    total credits += subject[i].credits;
            }
            SGPA = (double) effectiveScore / total credits;
    }
    Public void displayresult(){
            System.out.print("\n Student name:"+ name);
            System.out.print ("Student USN :" + usn);
            System.out.print (" SGPA:" + SGPA);
    }
}
public class Main{
        public static void main (String[] args){
                Student student = new Student();
                Student.getStudent Details();
                Student.get marks ();
                Student.compute SGPA();
                Student.display Result();
    }   }
}
```

OUTPUT

```
Enter Student Name:
Aman
Enter Student USN:
123
Enter marks for subject 1:
90
Enter credits for subject 1:
4
```



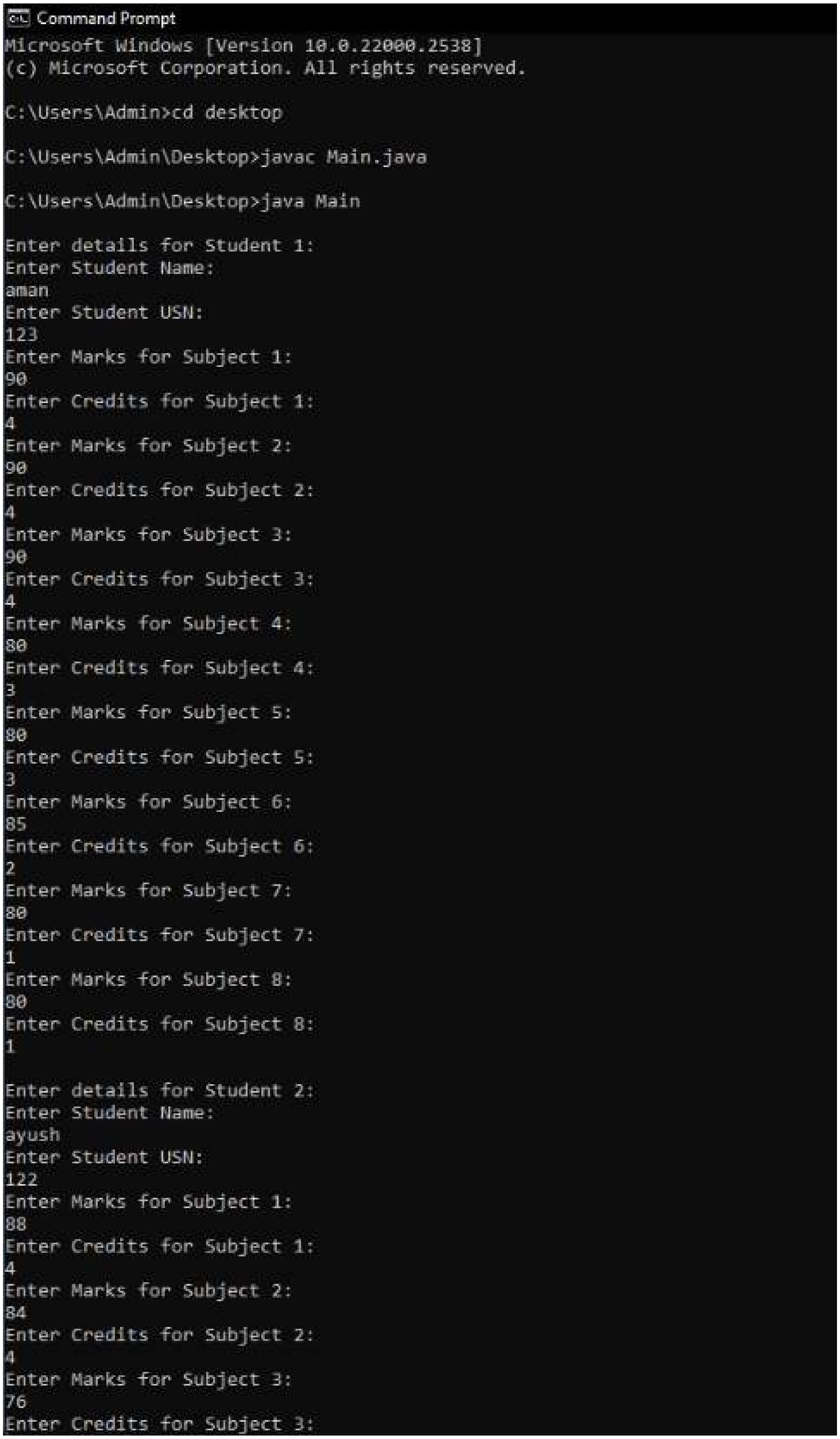
```
Enter marks for subject 2:
89
Enter credits for subject 2:
4
Enter marks for subject 3:
78
Enter credits for subject 3:
3
Enter marks for subject 4:
69
Enter credits for subject 5:
3
Enter marks for subject 5:
99
Enter credits for subject 5:
4
Enter marks for subject 6:
92
Enter credits for subject 6:
2
Enter marks for subject 7:
94
Enter credits for subject 7:
1
Enter marks for subject 8:
87
Enter credits for subject 8:
1

Student Name: Aman
Student USN: 123
SGPA: 9.090909092
```

16/10/24

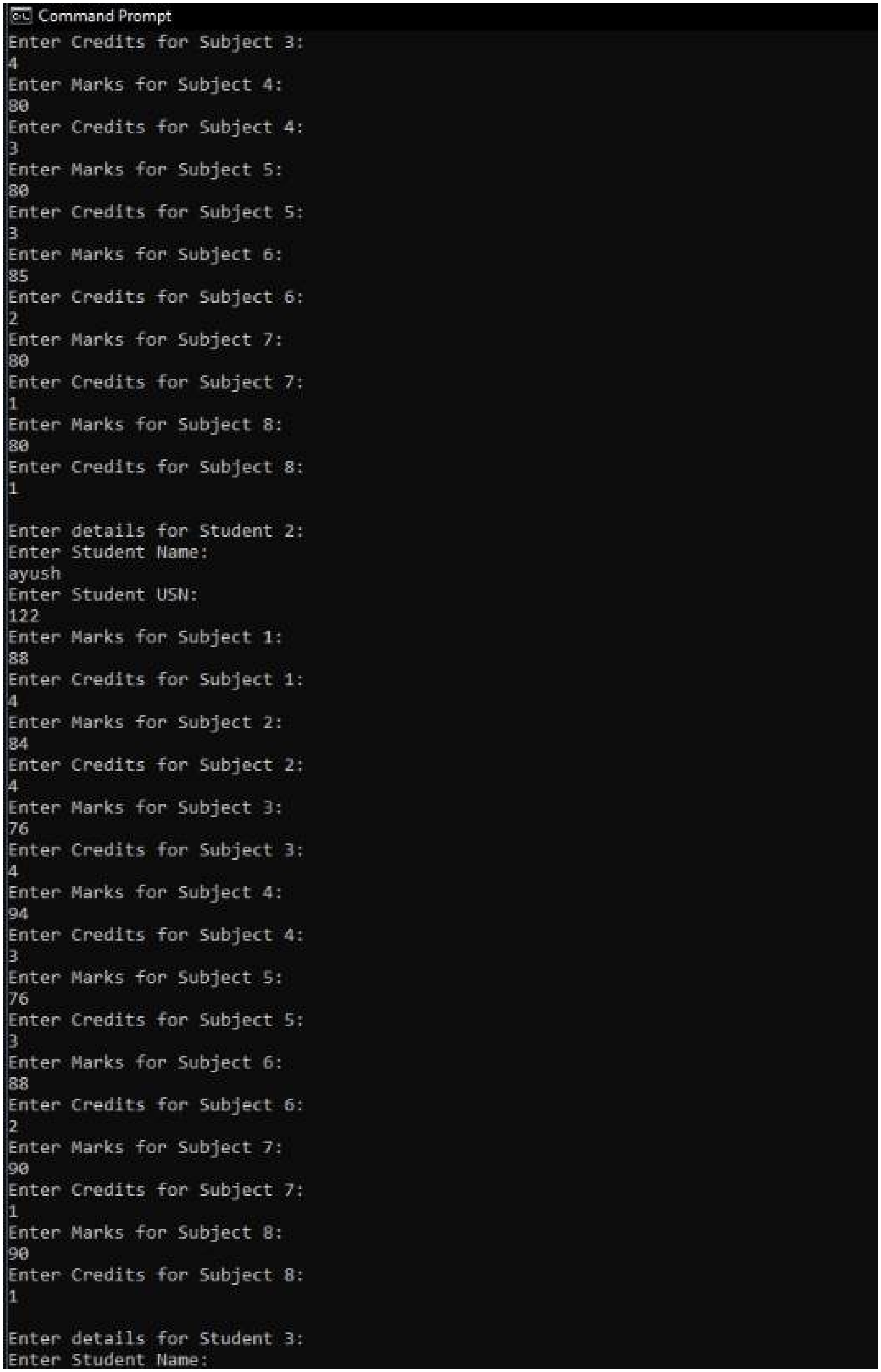
Code:

```
import java.util.Scanner;

class Subject {
    int subjectMarks;
    int credits;
    int grade;


    public void calculateGrade() {
        if (subjectMarks >= 90 && subjectMarks <= 100) {
            grade = 10;
        } else if (subjectMarks >= 80) {
            grade = 9;
        } else if (subjectMarks >= 70) {
            grade = 8;
        } else if (subjectMarks >= 60) {
            grade = 7;
        } else if (subjectMarks >= 50) {
            grade = 6;
        } else if (subjectMarks >= 40) {
            grade = 5;
        } else {
            grade = 0;
        }
    }
}

class Student {
    String name;
    String usn;
    double SGPA;
    Subject[] subject = new Subject[8];
    Scanner s = new Scanner(System.in);


    public Student() {
        for (int i = 0; i < 8; i++) {
            subject[i] = new Subject();
        }
    }


    public void getStudentDetails() {
        System.out.println("Enter Student Name: ");
        name = s.nextLine();
        System.out.println("Enter Student USN: ");
        usn = s.nextLine();
    }


    public void getMarks() {
        for (int i = 0; i < 8; i++) {
            System.out.println("Enter Marks for Subject " + (i + 1) + ": ");
            subject[i].subjectMarks = s.nextInt();


            if (subject[i].subjectMarks > 100 || subject[i].subjectMarks < 0) {
                System.out.println("Invalid marks! Please enter again.");
                i--;
                continue;
            }

            System.out.println("Enter Credits for Subject " + (i + 1) + ": ");
            subject[i].credits = s.nextInt();

            subject[i].calculateGrade();
        }
    }


    public void computeSGPA() {
        int totalCredits = 0;
        int effectiveScore = 0;

        for (int i = 0; i < 8; i++) {
            effectiveScore += (subject[i].grade * subject[i].credits);
            totalCredits += subject[i].credits;
        }

        SGPA = (double) effectiveScore / totalCredits;
    }


    public void displayResult() {
        System.out.println("\nStudent Name: " + name);
        System.out.println("Student USN: " + usn);
        System.out.println("SGPA: " + SGPA);
    }
}

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


        Student[] students = new Student[3];


        for (int i = 0; i < 3; i++) {
            System.out.println("\nEnter details for Student " + (i + 1) + ": ");
            students[i] = new Student();
            students[i].getStudentDetails();
            students[i].getMarks();
            students[i].computeSGPA();
        }


        System.out.println("\n\nResults for all students:");
        for (int i = 0; i < 3; i++) {
            students[i].displayResult();
        }
    }
}
```

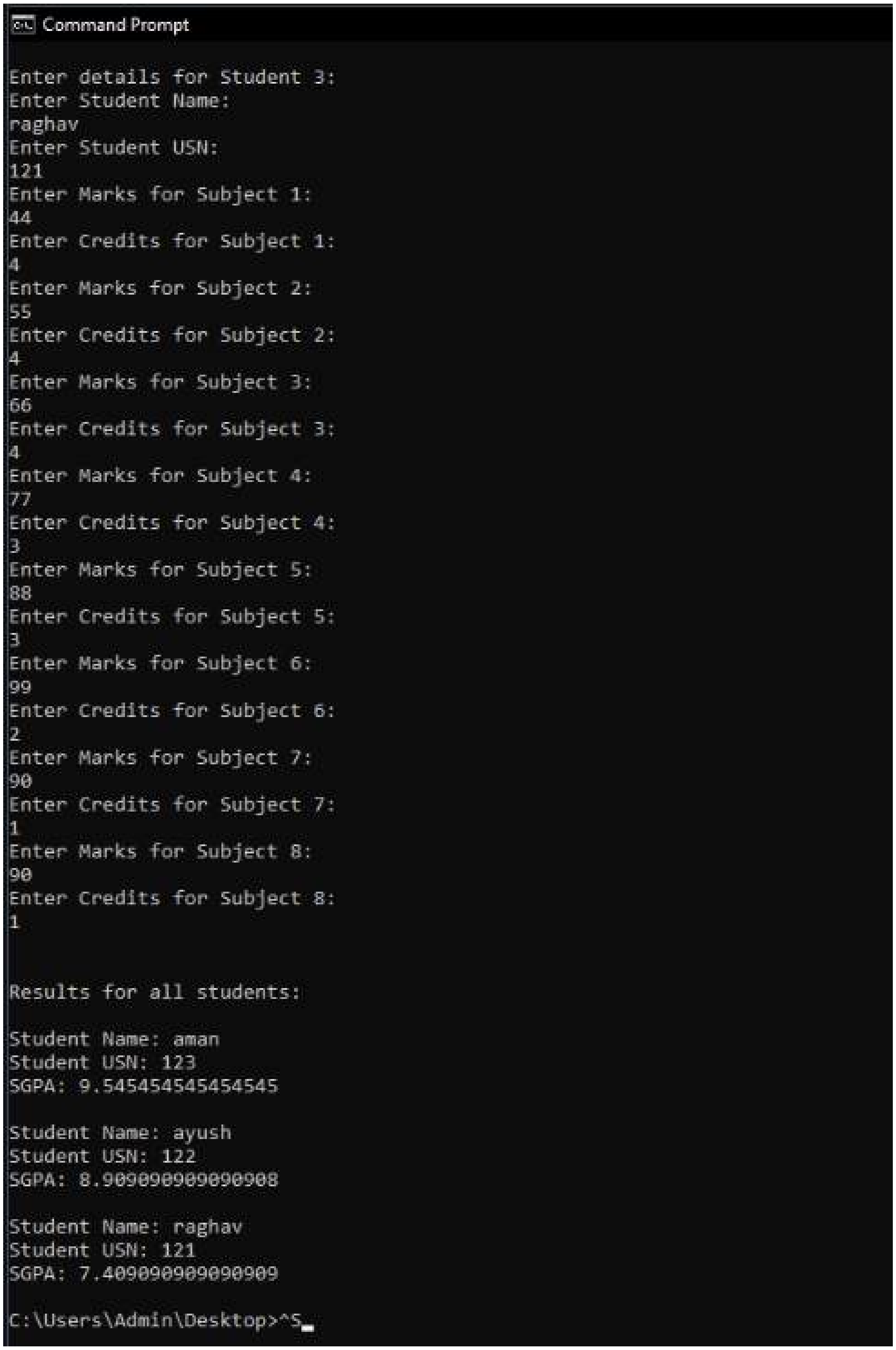
Output:



```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac Main.java

C:\Users\Admin\Desktop>java Main

Enter details for Student 1:
Enter Student Name:
aman
Enter Student USN:
123
Enter Marks for Subject 1:
90
Enter Credits for Subject 1:
4
Enter Marks for Subject 2:
90
Enter Credits for Subject 2:
4
Enter Marks for Subject 3:
90
Enter Credits for Subject 3:
4
Enter Marks for Subject 4:
80
Enter Credits for Subject 4:
3
Enter Marks for Subject 5:
80
Enter Credits for Subject 5:
3
Enter Marks for Subject 6:
85
Enter Credits for Subject 6:
2
Enter Marks for Subject 7:
80
Enter Credits for Subject 7:
1
Enter Marks for Subject 8:
80
Enter Credits for Subject 8:
1

Enter details for Student 2:
Enter Student Name:
ayush
Enter Student USN:
122
Enter Marks for Subject 1:
88
Enter Credits for Subject 1:
4
Enter Marks for Subject 2:
84
Enter Credits for Subject 2:
4
Enter Marks for Subject 3:
76
Enter Credits for Subject 3:
```



```
Enter Credits for Subject 3:
4
Enter Marks for Subject 4:
80
Enter Credits for Subject 4:
3
Enter Marks for Subject 5:
80
Enter Credits for Subject 5:
3
Enter Marks for Subject 6:
85
Enter Credits for Subject 6:
2
Enter Marks for Subject 7:
80
Enter Credits for Subject 7:
1
Enter Marks for Subject 8:
80
Enter Credits for Subject 8:
1

Enter details for Student 2:
Enter Student Name:
ayush
Enter Student USN:
122
Enter Marks for Subject 1:
88
Enter Credits for Subject 1:
4
Enter Marks for Subject 2:
84
Enter Credits for Subject 2:
4
Enter Marks for Subject 3:
76
Enter Credits for Subject 3:
4
Enter Marks for Subject 4:
94
Enter Credits for Subject 4:
3
Enter Marks for Subject 5:
76
Enter Credits for Subject 5:
3
Enter Marks for Subject 6:
88
Enter Credits for Subject 6:
2
Enter Marks for Subject 7:
90
Enter Credits for Subject 7:
1
Enter Marks for Subject 8:
90
Enter Credits for Subject 8:
1

Enter details for Student 3:
Enter Student Name:
```

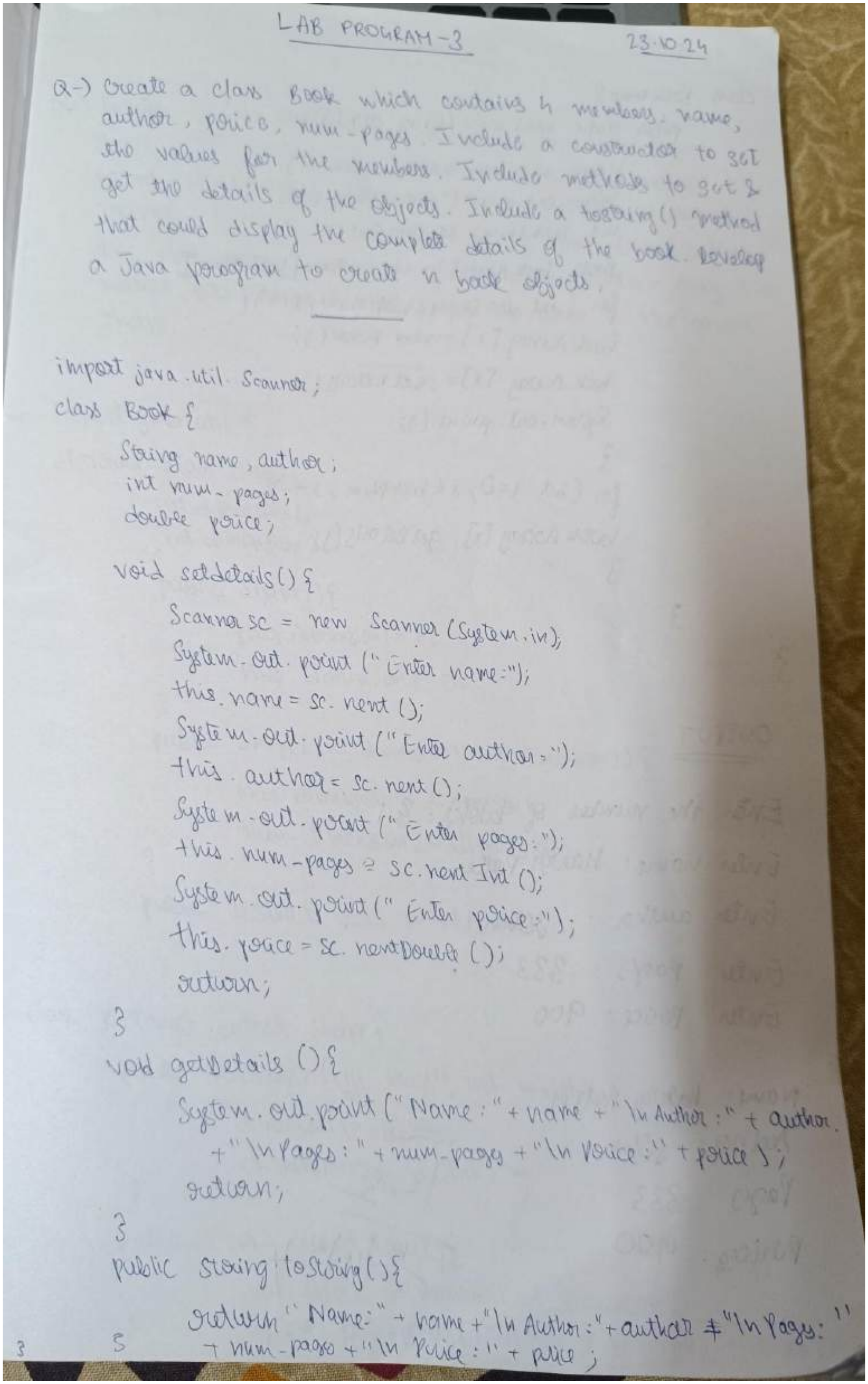


```
Enter details for Student 3:
Enter Student Name:
raghav
Enter Student USN:
121
Enter Marks for Subject 1:
44
Enter Credits for Subject 1:
4
Enter Marks for Subject 2:
55
Enter Credits for Subject 2:
4
Enter Marks for Subject 3:
66
Enter Credits for Subject 3:
4
Enter Marks for Subject 4:
77
Enter Credits for Subject 4:
3
Enter Marks for Subject 5:
88
Enter Credits for Subject 5:
3
Enter Marks for Subject 6:
99
Enter Credits for Subject 6:
2
Enter Marks for Subject 7:
90
Enter Credits for Subject 7:
1
Enter Marks for Subject 8:
90
Enter Credits for Subject 8:
1


Results for all students:

Student Name: aman
Student USN: 123
SGPA: 9.545454545454545

Student Name: ayush
Student USN: 122
SGPA: 8.909090909090908

Student Name: raghav
Student USN: 121
SGPA: 7.409090909090909

C:\Users\Admin\Desktop>^S
```

Program-3

BOOK CLASS

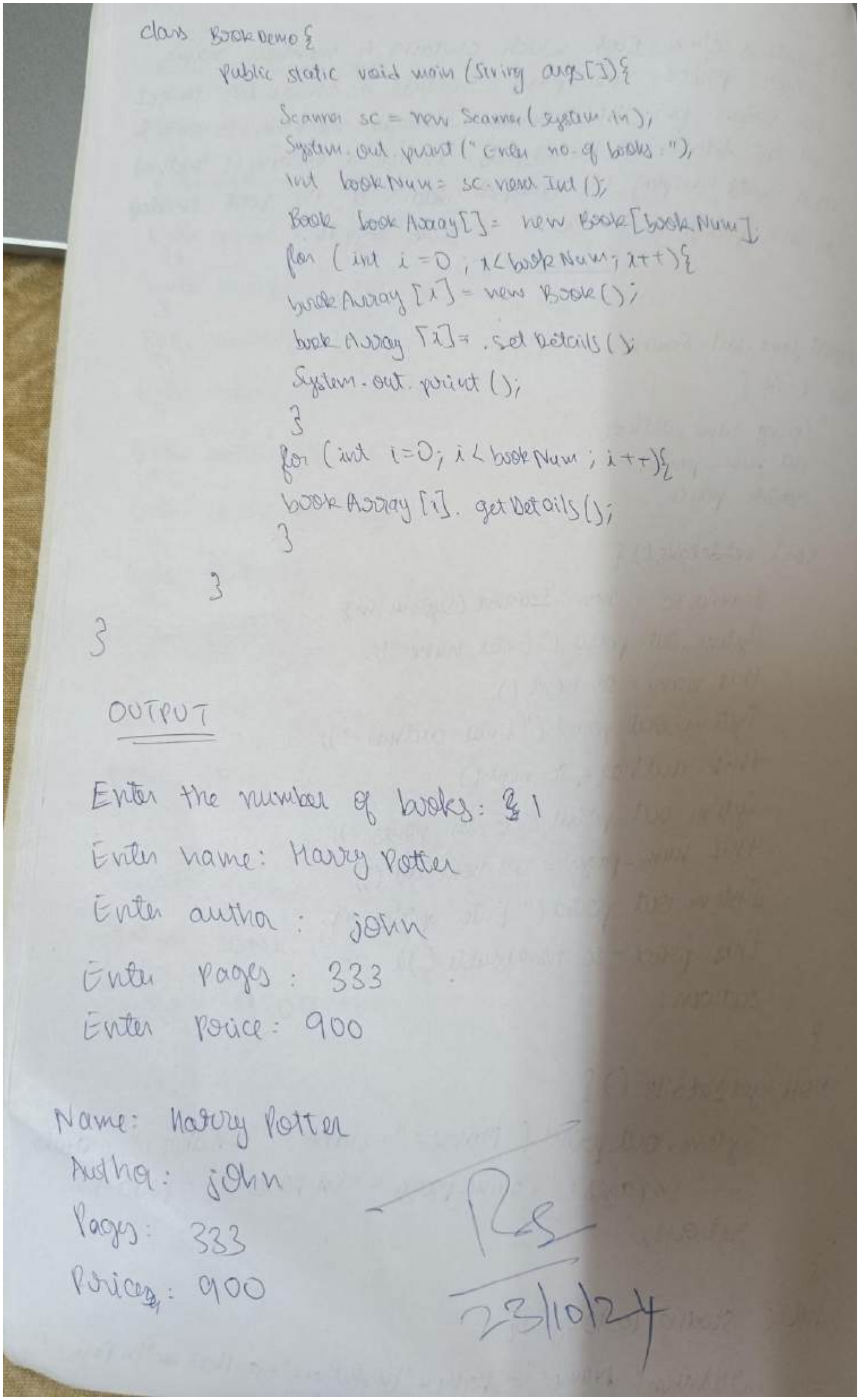
Observation:



LAB PROGRAM-3 23.10.24

Q-) Create a class Book which contains 4 members: name, author, price, num-pages. Include a constructor to set the values for the members. Include methods to set & 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 {
    String name, author;
    int num-pages;
    double price;

    void setdetails(){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter name:");
        this.name = sc.next();
        System.out.print("Enter author:");
        this.author = sc.next();
        System.out.print("Enter pages:");
        this.num-pages = sc.nextInt();
        System.out.print("Enter price:");
        this.price = sc.nextDouble();
        return;
    }
    void getDetails(){
        System.out.print("Name:"+name+"\n Author:"+author
          +"\nPages:"+num-pages+"\n Price:"+price);
        return;
    }
    public String toString(){
        return "Name:"+name+"\n Author:"+author +"\n Pages:"
}   }   + num-pages+"\n Price:"+price;
```



```
class BookDemo{
    public static void main (String args[]){
        Scanner sc = new Scanner( system.in);
        System.out.print("Enter no. of books: "),
        int bookNum= sc.nextInt();
        Book bookArray[]= new Book[bookNum].
        for (int i =0 , i<bookNum; i++){
        bookArray [i] = new Book();
        bookArray [i]= .setDetails();
        System.out.print();
        }
        for (int i=0; i<bookNum ; i++){
        bookArray [i]. getDetails();
        }
    }
}
```

OUTPUT

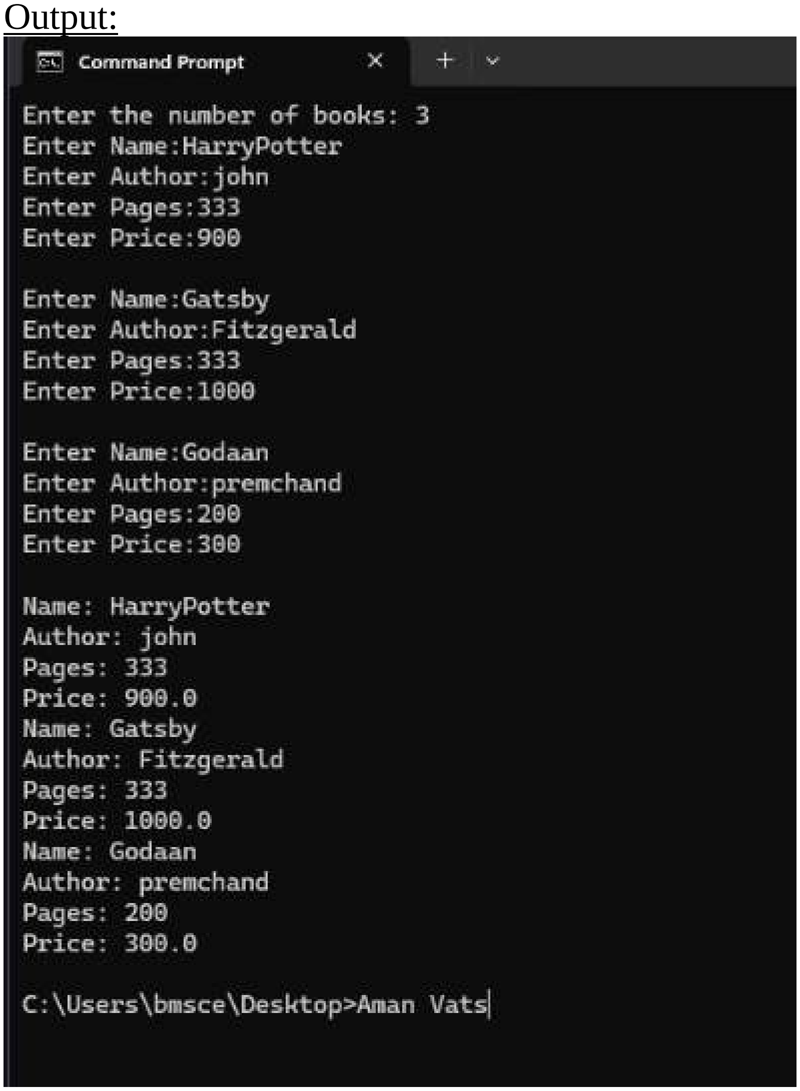
```
Enter the number of books: 1
Enter name: Harry Potter
Enter author : john
Enter pages : 333
Enter price: 900

Name: Harry Potter
Author: john
Pages: 333
Price: 900
```

23/10/24

Code:

```
import java.util.Scanner;
class Book{
    String name, author;
    int num_pages;
    double price;

    void setDetails(){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Name:");
        this.name = sc.next();
        System.out.print("Enter Author:");
        this.author = sc.next();
        System.out.print("Enter Pages:");
        this.num_pages = sc.nextInt();
        System.out.print("Enter Price:");
        this.price = sc.nextDouble();
        return;
    }

    void getDetails(){
        System.out.println("Name: "+name+"\nAuthor: "+author+"\nPages: "+num_pages+"\nPrice: 
"+price);
        return;
    }

    public String toString(){
        return "Name: "+name+"\nAuthor: "+author+"\nPages: "+num_pages+"\nPrice: "+price;
    }
}

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

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        int bookNum = sc.nextInt();

        Book bookArray[] = new Book[bookNum];

        for(int i = 0; i<bookNum; i++){
        bookArray[i] = new Book();
        bookArray[i].setDetails();
        System.out.println();
        }
        for (int i = 0; i<bookNum; i++){
        bookArray[i].getDetails();
        }
    }
}
```

Output:

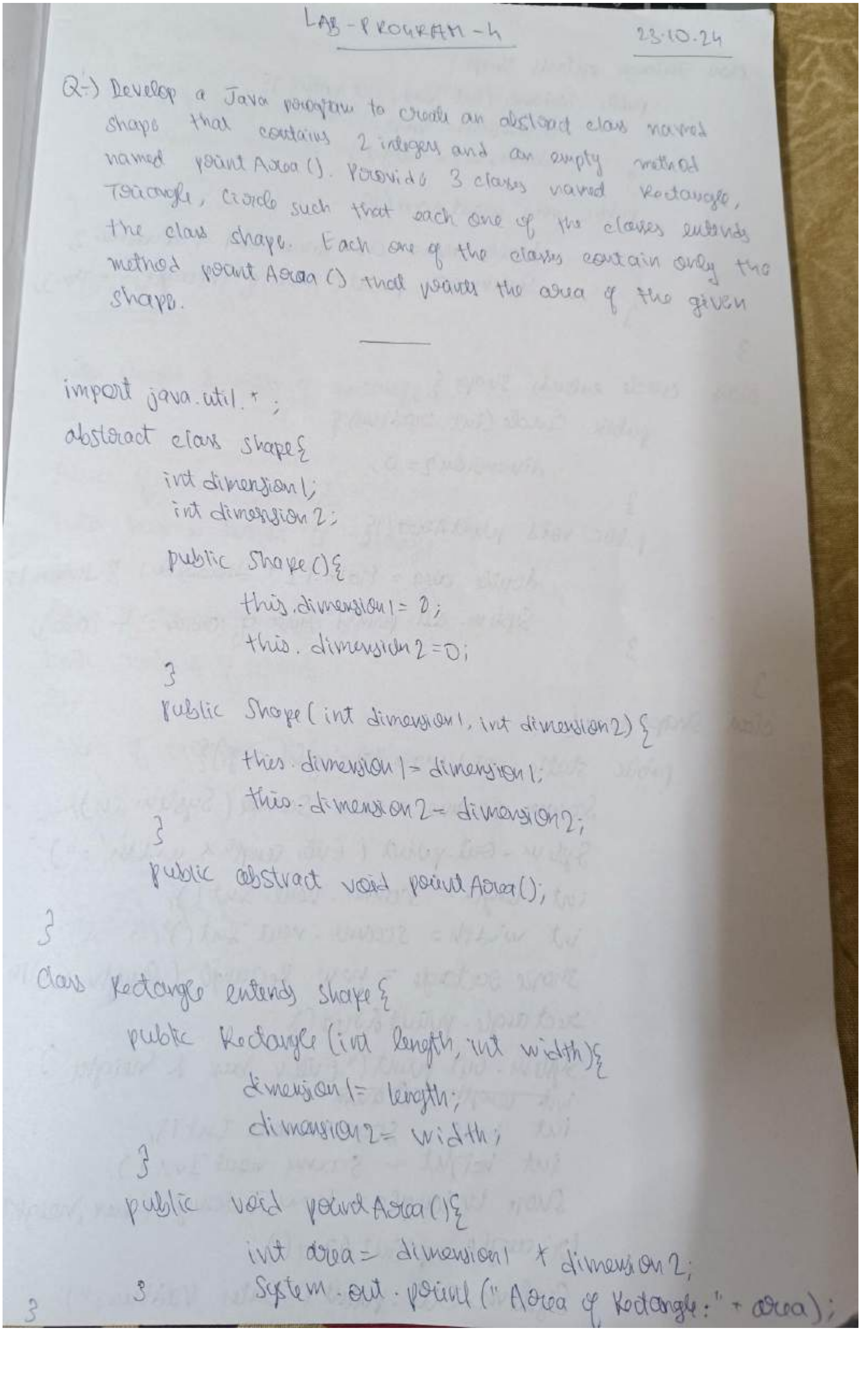


```
Enter the number of books: 3
Enter Name:HarryPotter
Enter Author:john
Enter Pages:333
Enter Price:900

Enter Name:Gatsby
Enter Author:Fitzgerald
Enter Pages:333
Enter Price:1000

Enter Name:Godaan
Enter Author:premchand
Enter Pages:200
Enter Price:300

Name: HarryPotter
Author: john
Pages: 333
Price: 900.0
Name: Gatsby
Author: Fitzgerald
Pages: 333
Price: 1000.0
Name: Godaan
Author: premchand
Pages: 200
Price: 300.0

C:\Users\bmsce\Desktop>Aman Vats
```

Program-4

Abstract Class Shape

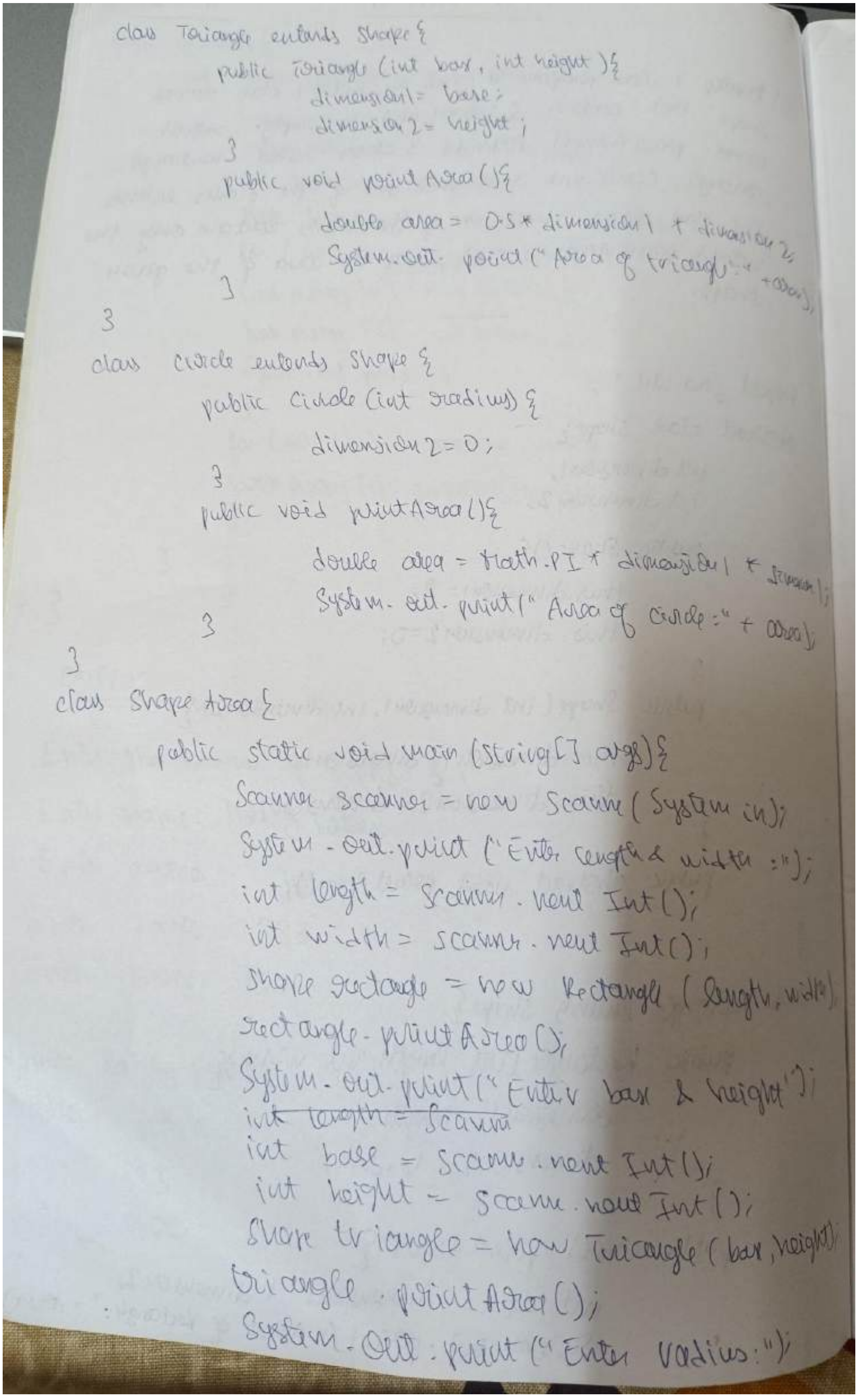
Observation:



LAB - PROGRAM - 4 23.10.24

Q-) Develop a Java program to create an abstract class named Shape that contains 2 integers and an empty method named printArea(). Provide 3 classes named Rectangle, Triangle, 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.*;
abstract class shape{
        int dimension1;
        int dimension2;
        public Shape(){
                this.dimension1= 0;
                this.dimension2=0;
        }
        public Shape(int dimension1, int dimension2){
                this.dimension1= dimension1;
                this.dimension2= dimension2;
        }
        public abstract void printArea();
}
Class Rectangle extends shape{
    public Rectangle(int length, int width){
            dimension1= length;
            dimension2= width;
    }
    public void printArea(){
            int area= dimension1 * dimension2;
            System.out.print("Area of Rectangle:" + area);
    }
}
```



```
class Triangle extends Shape {
        public Triangle (int base, int height ){
                dimension1= base;
                dimension2= height ;
        }
        public void printArea(){
                double area = 0.5 * dimension1 * dimension2;
                System.out. print (" Area of triangle:" +area);
        }
}
class circle extends Shape {
        public Circle (int radius) {
                dimension2=0;
        }
        public void printArea(){
                double area = Math.PI * dimension1 * dimension1;
                System. out. print (" Area of circle :" + area);
        }
}
class ShapeArea {
        public static void main (String[] args){
                Scanner scanner = new Scanner ( System in);
                System - out.print (" Enter length & width :");
                int length = Scanner . next Int ();
                int width = scanner . next Int();
                Shape rectangle = new Rectangle ( length, width)
                rectangle. printArea ();
                System. out. print (" Enter base & height");
                ~~int length = Scanne~~
                int base = Scanne . next Int ();
                int height = Scanne . next Int ();
                Shape triangle = new Triangle ( base, height)
                triangle . print Area ();
                System. Out . print (" Enter radius:");
        int radius = Scanner.nextInt();
        Shape circle = new Circle(radius);
        circle.printArea();
        Scanner.close();
    }
}
```

OUTPUT

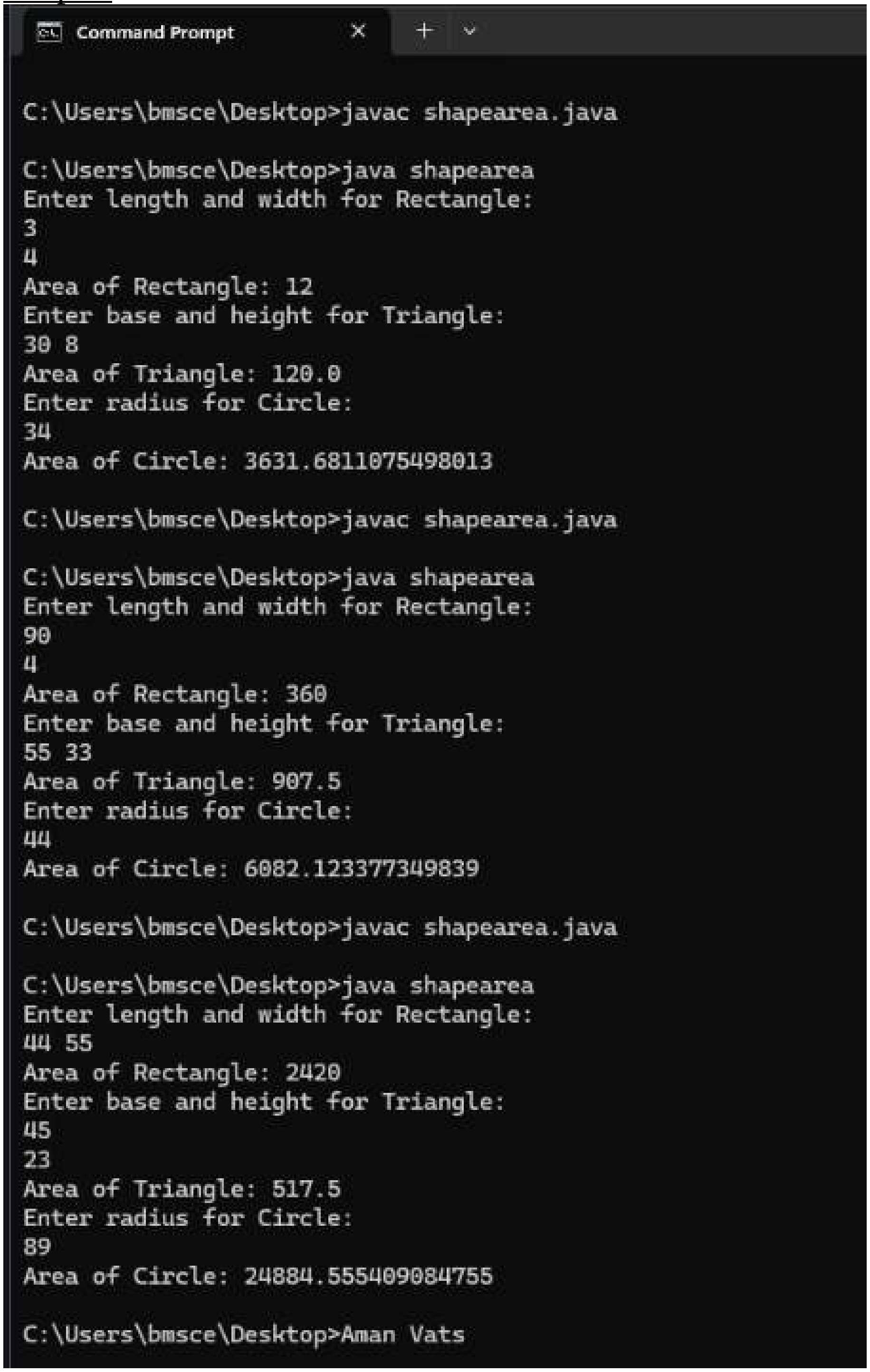
```
Enter length & width of rectangle:
3
4
Area of rectangle : 12
Enter base & height of triangle:
30 8
Area of triangle: 120.0
Enter radius of circle:
34
Area of circle: 3631.6811
```

23/10/24

Code:

```
import java.util.*;

abstract class Shape {
    int dimension1;
    int dimension2;


    public Shape() {
        this.dimension1 = 0;
        this.dimension2 = 0;

    }

    public Shape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }

    public abstract void printArea();

}

class Rectangle extends Shape {
    public Rectangle(int length, int width) {

        dimension1 = length;
        dimension2 = width;
    }

    public void printArea() {

        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);

    }
}

class Triangle extends Shape {
    public Triangle(int base, int height) {

        dimension1 = base;
        dimension2 = height;
    }

    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);

    }
}

class Circle extends Shape {
    public Circle(int radius) {

        dimension1 = radius;
        dimension2 = 0;
    }

    public void printArea() {

        double area = Math.PI * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

 class shapearea {

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

        System.out.println("Enter length and width for Rectangle:");

        int length = scanner.nextInt();
        int width = scanner.nextInt();
        Shape rectangle = new Rectangle(length, width);
        rectangle.printArea();

        System.out.println("Enter base and height for Triangle:");

        int base = scanner.nextInt();
        int height = scanner.nextInt();
        Shape triangle = new Triangle(base, height);
        triangle.printArea();

        System.out.println("Enter radius for Circle:");

        int radius = scanner.nextInt();
        Shape circle = new Circle(radius);
        circle.printArea();

        scanner.close();
    }
}
```

Output:

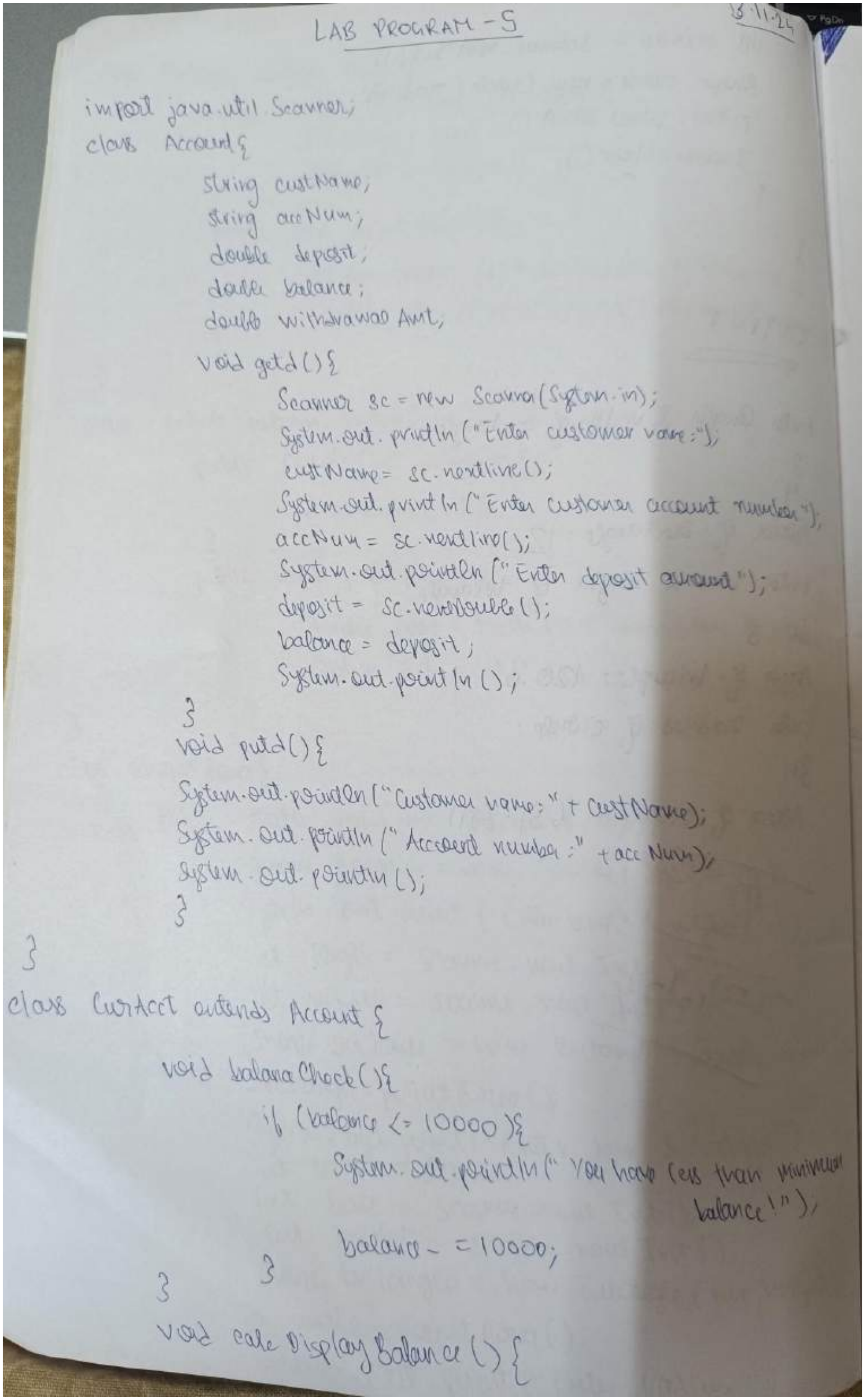


```
C:\Users\bmsce\Desktop>javac shapearea.java

C:\Users\bmsce\Desktop>java shapearea
Enter length and width for Rectangle:
3
4
Area of Rectangle: 12
Enter base and height for Triangle:
30 8
Area of Triangle: 120.0
Enter radius for Circle:
34
Area of Circle: 3631.6811075498013

C:\Users\bmsce\Desktop>javac shapearea.java

C:\Users\bmsce\Desktop>java shapearea
Enter length and width for Rectangle:
90
4
Area of Rectangle: 360
Enter base and height for Triangle:
55 33
Area of Triangle: 907.5
Enter radius for Circle:
44
Area of Circle: 6082.123377349839

C:\Users\bmsce\Desktop>javac shapearea.java

C:\Users\bmsce\Desktop>java shapearea
Enter length and width for Rectangle:
44 55
Area of Rectangle: 2420
Enter base and height for Triangle:
45
23
Area of Triangle: 517.5
Enter radius for Circle:
89
Area of Circle: 24884.555409084755

C:\Users\bmsce\Desktop>Aman Vats
```

Program-5

BANK CLASS

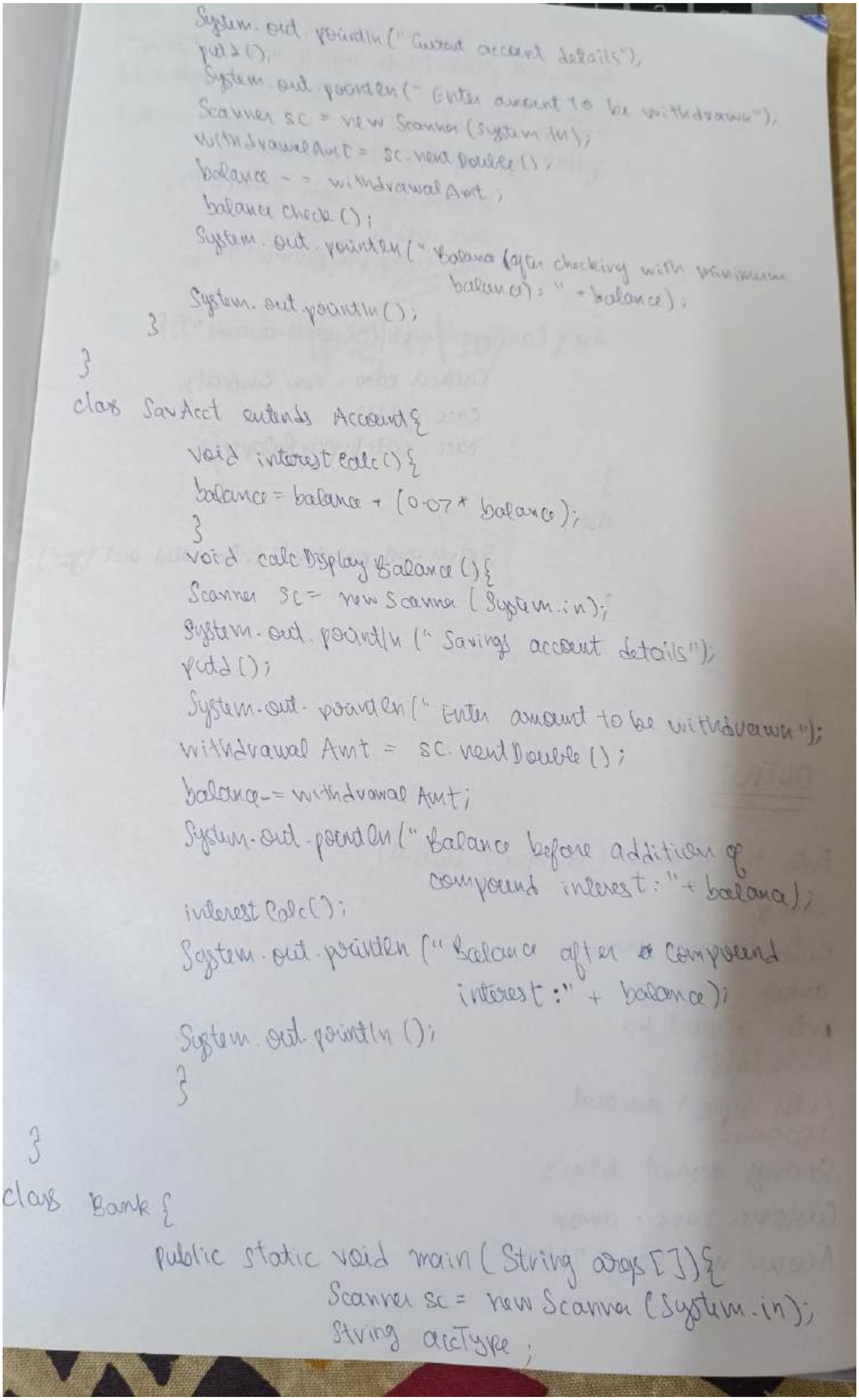
Observation:



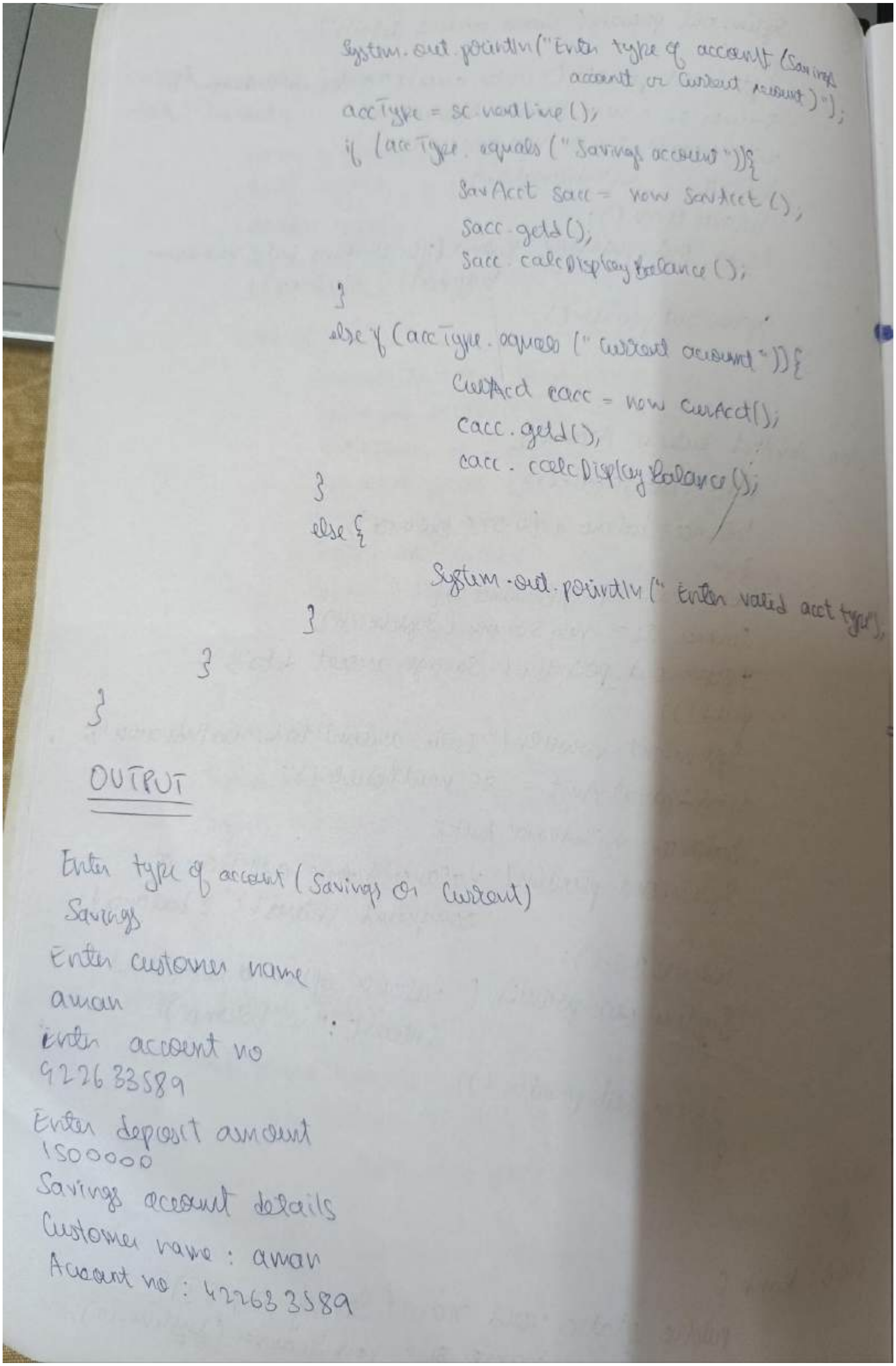
LAB PROGRAM - 5

13.11.24

```
import java.util.Scanner;
class Account{
        String custName;
        String accNum;
        double deposit;
        double balance;
        double withdrawal Amt;
       void getd(){
              Scanner sc = new Scanner(System.in);
              System.out.println("Enter customer name:");
              custName= sc.nextline();
              System.out.println("Enter customer account number");
              accNum= sc.nextline();
              System.out.pointln("Enter deposit amount");
              deposit = sc.nextDouble();
              balance = deposit;
              System.out.point ln();
      }
      void putd(){
      System.out.pointln("Customer name:" + custName);
      System.out.pointln("Account number:" +acc Num);
      System.out.pointln();
      }
}
class CurAcct extends Account {
        void balanceCheck(){
                if (balance <= 10000 ){
                        System.out.pointln(" You have less than minimum
                                                              balance!");
                        balance - =10000;
                }
        }
        void calc Display Balance () {
```



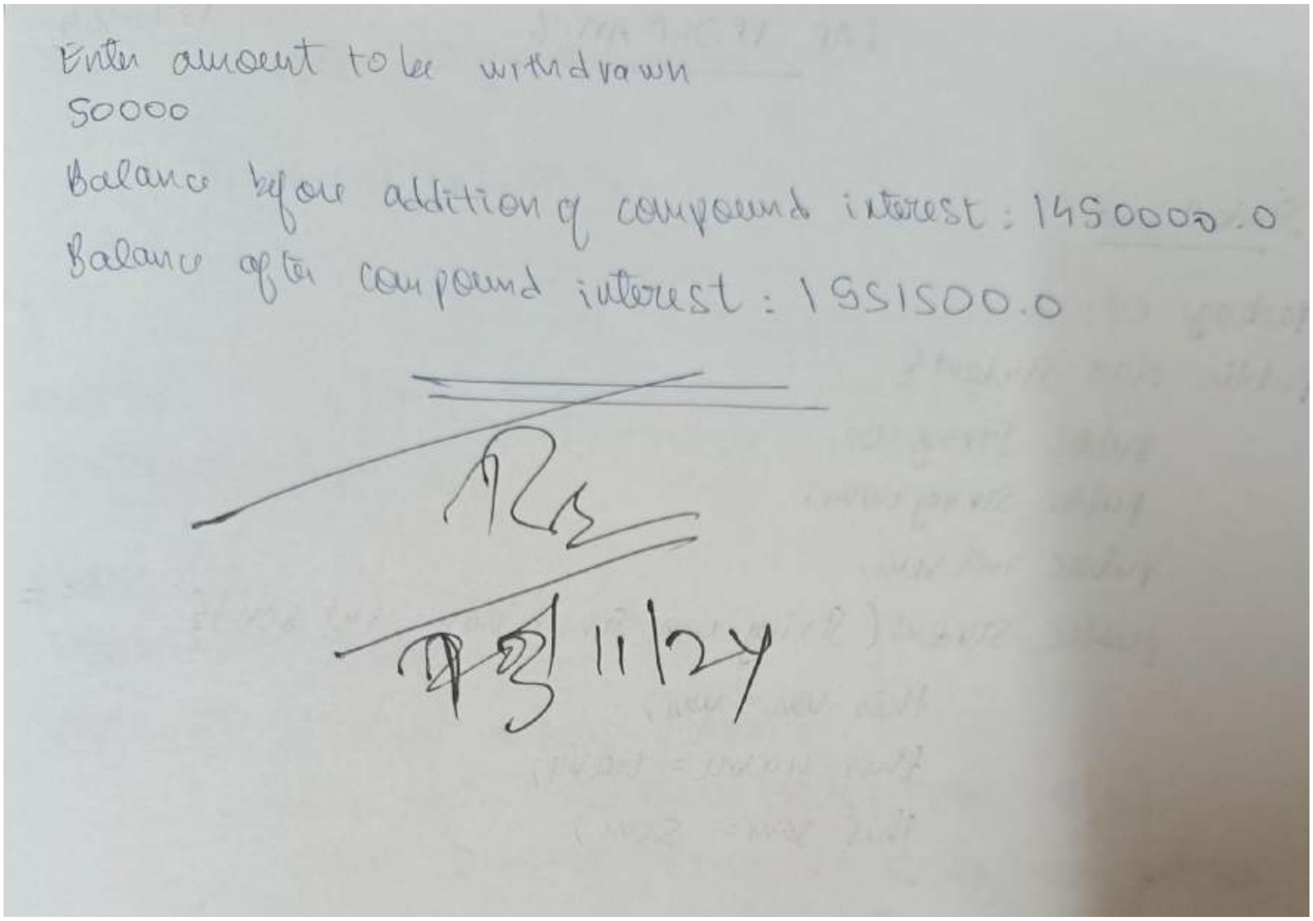
```
        System.out.println("Current account details");
        putd();
        System.out.println("Enter amount to be withdrawn");
        Scanner sc = new Scanner(System.in);
        withdrawalAmt = sc.nextDouble();
        balance -= withdrawalAmt;
        balance check();
        System.out.println("Balance (after checking with minimum
                                balance): " + balance);
        System.out.println();
    }
}
class SavAcct extends Account{
        void interestCalc(){
        balance = balance + (0.07* balance);
        }
        void calcDisplayBalance(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Savings account details");
        putd();
        System.out.println("Enter amount to be withdrawn");
        withdrawal Amt = sc.nextDouble();
        balance -= withdrawal Amt;
        System.out.println("Balance before addition of
                        compound interest: " + balance);
        interest Calc();
        System.out.println("Balance after compound
                            interest:" + balance);
        System.out.println();
        }
}
class Bank{
    public static void main(String args[]){
                Scanner sc = new Scanner(System.in);
                String accType;
```



```
          System.out.println("Enter type of account (Savings
                                   account or Current account)");
          accType = sc.nextLine();
          if (accType.equals("Savings account")){
                    SavAcct sacc = new SavAcct();
                    sacc.getd();
                    sacc.calcDisplayBalance();
          }
          else if (accType.equals("Current account")){
                    CurAcct cacc = new CurAcct();
                    cacc.getd(),
                    cacc.calcDisplayBalance();
          }
          else {
                    System.out.println("Enter valid acct type");
          }
     }
}
```

OUTPUT

```
Enter type of account (Savings or Current)
Savings
Enter customer name
aman
Enter account no
922633589
Enter deposit amount
1500000
Savings account details
Customer name : aman
Account no : 422633589
```



```
Enter amount to be withdrawn
50000
Balance before addition of compound interest : 1450000.0
Balance after compound interest : 1551500.0
```

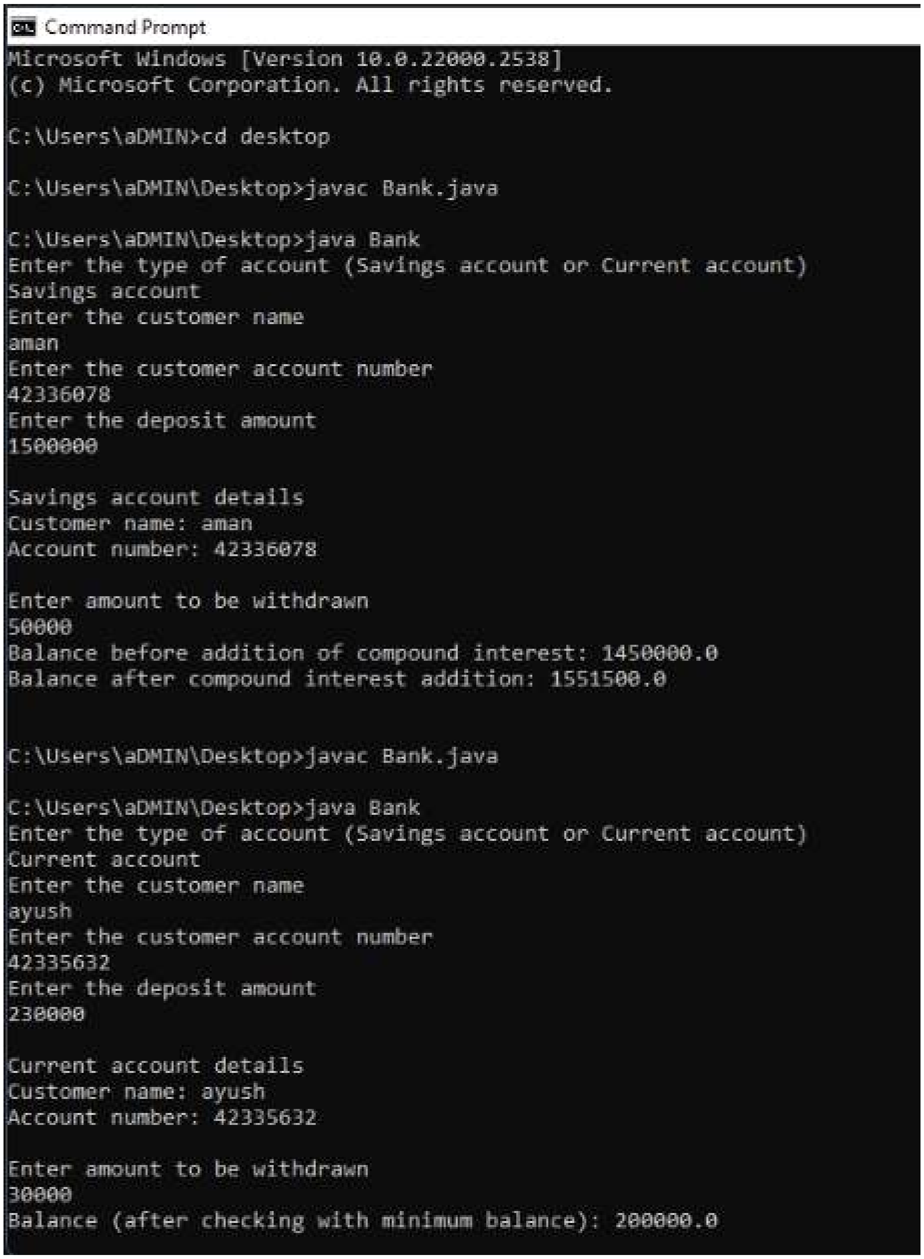
13/11/24

Code:

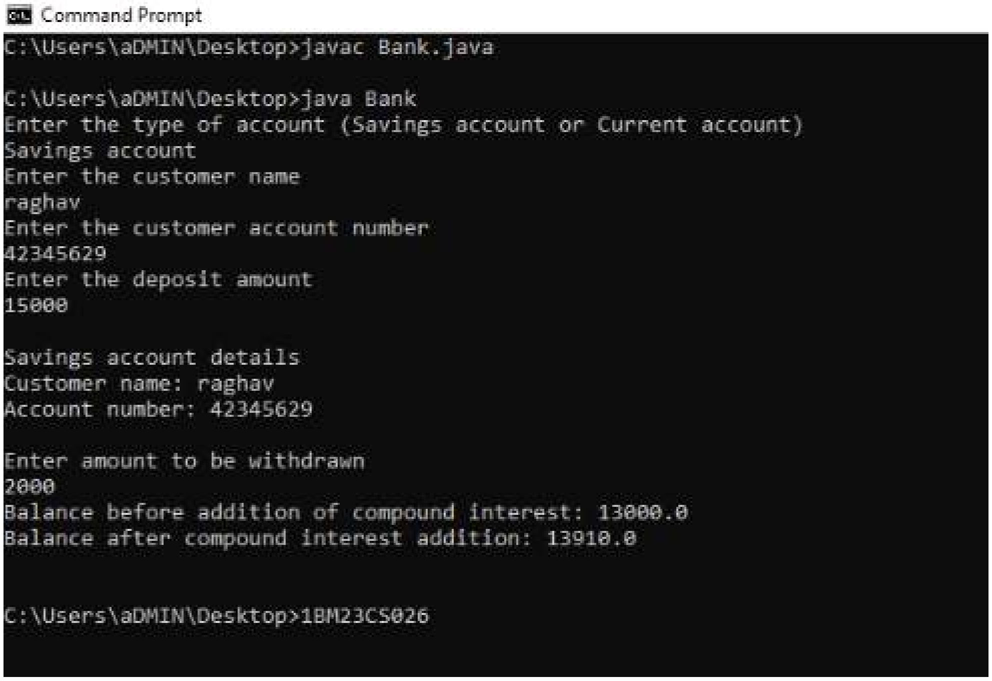
```
import java.util.Scanner;
class Account {
        String custName;
        String accNum;
        double deposit;
        double balance;
        double withdrawalAmt;

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

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

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

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

class Bank {
	public static void main(String args[]) {
		Scanner sc = new Scanner(System.in);
		String accType;
		System.out.println("Enter the type of account (Savings account or Current account)");
		accType = sc.nextLine();
		if (accType.equals("Savings account")) {
			SavAcct sacc = new SavAcct();
			sacc.getd();
			sacc.calcDisplayBalance();
		}
		else if (accType.equals("Current account")) {
			CurAcct cacc = new CurAcct();
			cacc.getd();
			cacc.calcDisplayBalance();
		}
		else {
			System.out.println("Enter a valid account type");
		}
	}
}
```

Output:



```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\aDMIN>cd desktop

C:\Users\aDMIN\Desktop>javac Bank.java

C:\Users\aDMIN\Desktop>java Bank
Enter the type of account (Savings account or Current account)
Savings account
Enter the customer name
aman
Enter the customer account number
42336078
Enter the deposit amount
1500000

Savings account details
Customer name: aman
Account number: 42336078

Enter amount to be withdrawn
50000
Balance before addition of compound interest: 1450000.0
Balance after compound interest addition: 1551500.0


C:\Users\aDMIN\Desktop>javac Bank.java

C:\Users\aDMIN\Desktop>java Bank
Enter the type of account (Savings account or Current account)
Current account
Enter the customer name
ayush
Enter the customer account number
42335632
Enter the deposit amount
230000

Current account details
Customer name: ayush
Account number: 42335632

Enter amount to be withdrawn
30000
Balance (after checking with minimum balance): 200000.0
```

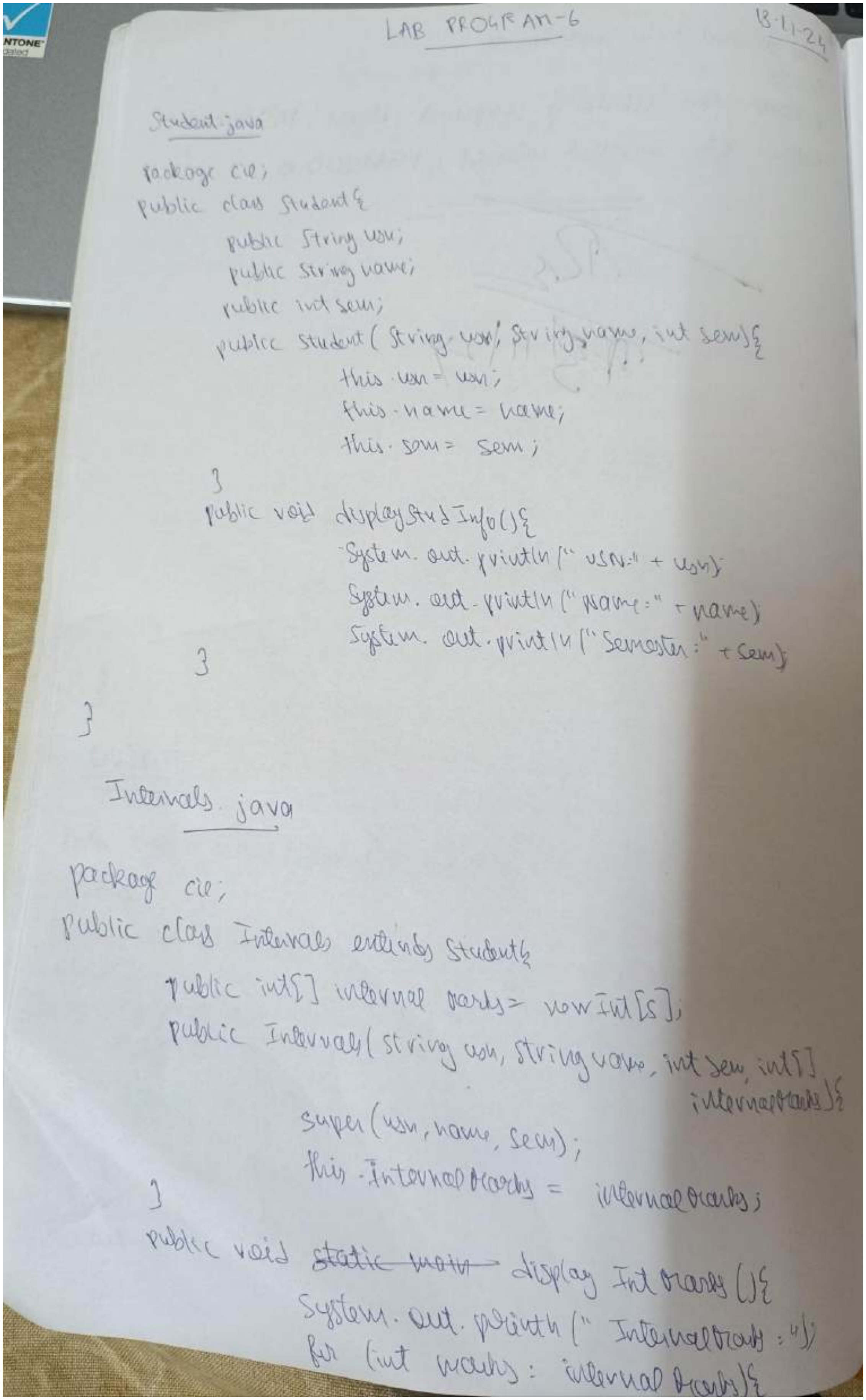


```
C:\Users\aDMIN\Desktop>javac Bank.java

C:\Users\aDMIN\Desktop>java Bank
Enter the type of account (Savings account or Current account)
Savings account
Enter the customer name
raghav
Enter the customer account number
42345629
Enter the deposit amount
15000

Savings account details
Customer name: raghav
Account number: 42345629

Enter amount to be withdrawn
2000
Balance before addition of compound interest: 13000.0
Balance after compound interest addition: 13910.0


C:\Users\aDMIN\Desktop>1BM23CS026
```

Program-6

Packages

Observation:



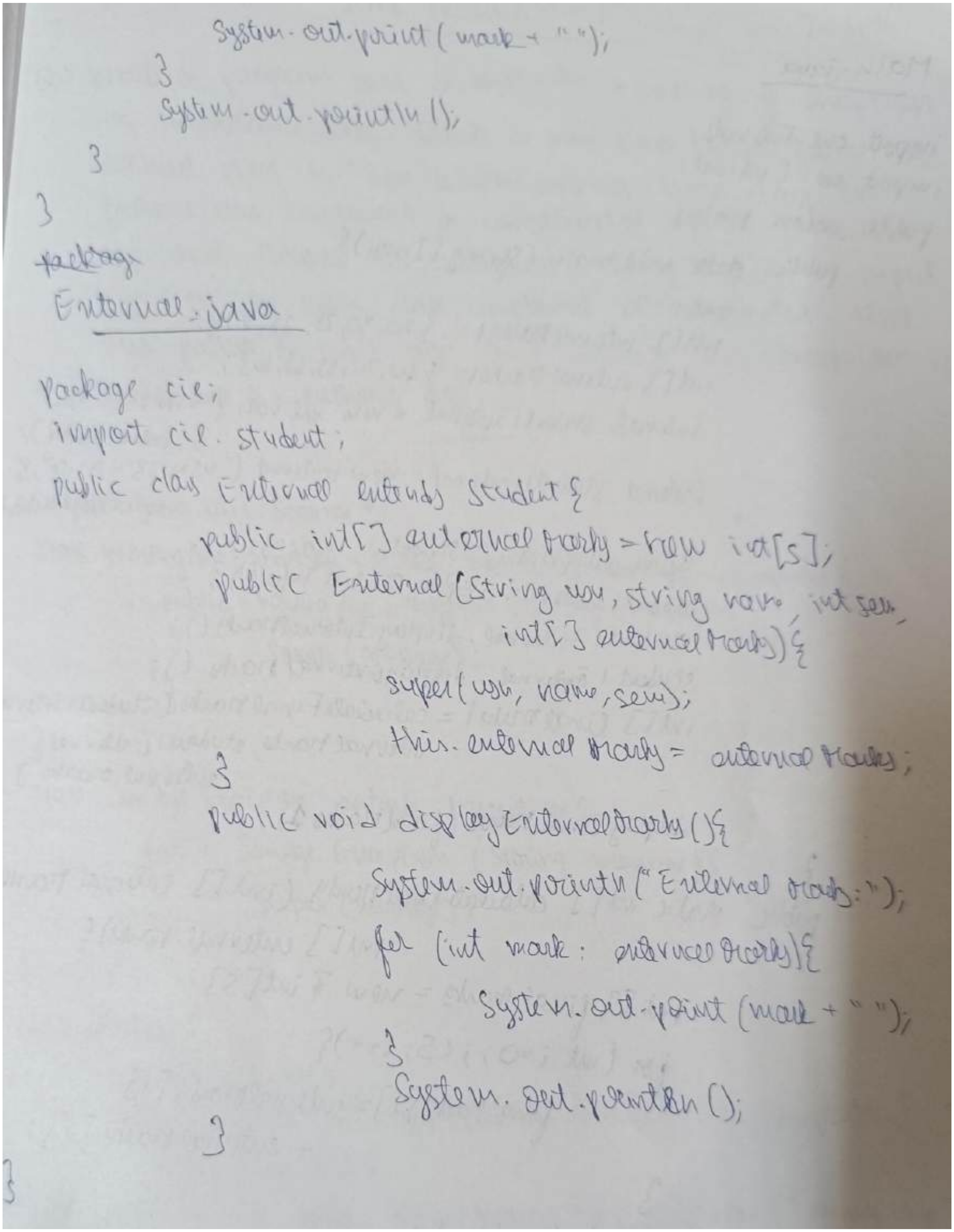
LAB PROGRAM-6 13-11-24

Student.java

```
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 displayStudInfo(){
                System.out.println("USN:" + usn);
                System.out.println("Name:" + name);
                System.out.println("Semester:" + sem);
        }
}
```

Internals.java

```
package cie;
public class Internals extends Student{
        public int[] internalMarks = new int[5];
        public Internals(String usn, String name, int sem, int[]
                                                    internalMarks){
                super(usn, name, sem);
                this.internalMarks = internalMarks;
        }
        public void ~~static main~~ displayIntMarks(){
                System.out.println("InternalMarks :");
                for (int marks : internalMarks){
```

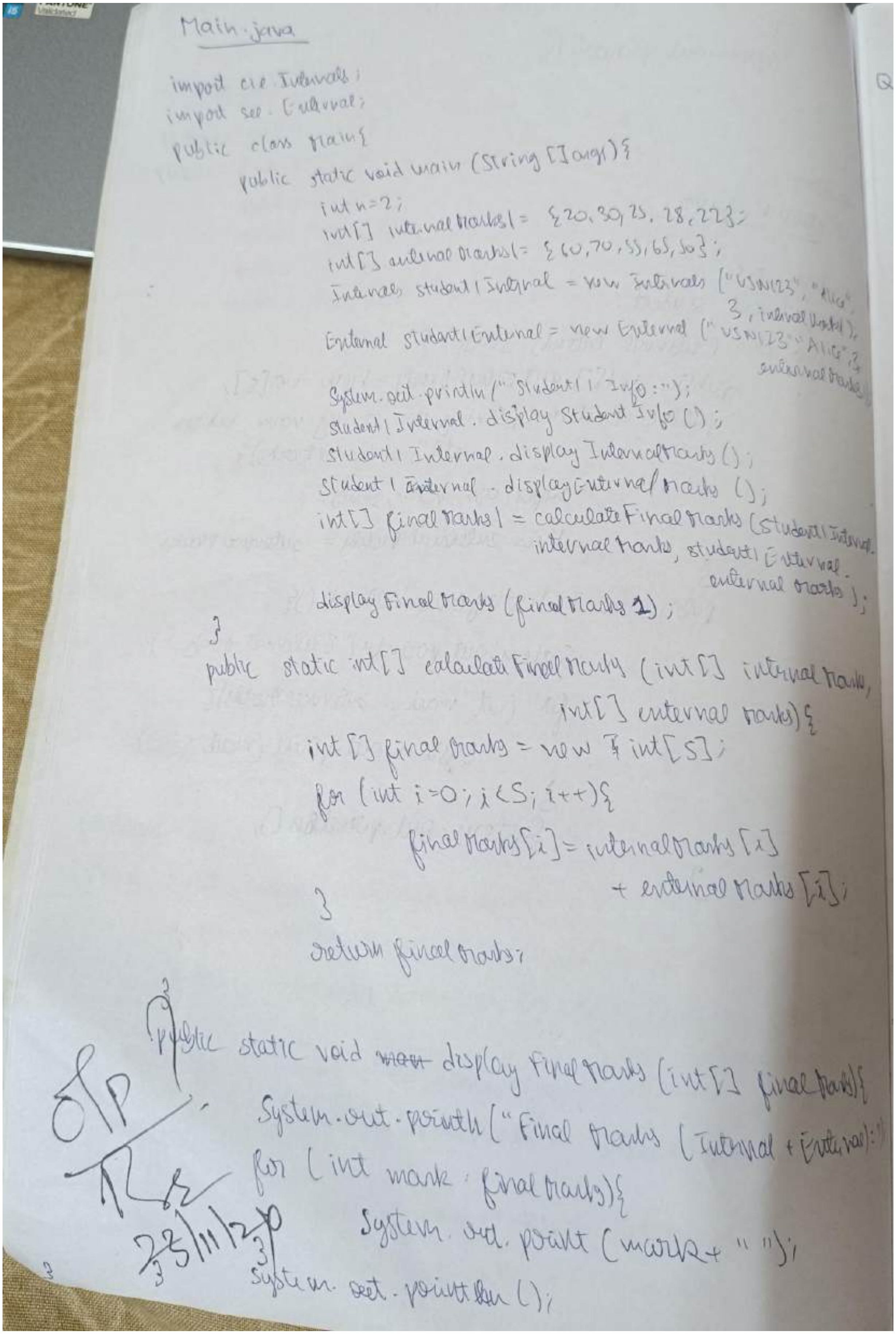


```
        System.out.print(mark + " ");
      }
      System.out.println();
   }
}
```

~~package~~

External.java

```
package cie;
import cie.student;
public class External extends Student{
        public int[] externalMarks = new int[5];
        public External(String usn, String name, int sem,
                        int[] externalMarks){
                super(usn, name, sem);
                this.externalMarks = externalMarks;
        }
        public void displayExternalMarks(){
                System.out.println("External marks:");
                for (int mark : externalMarks){
                        System.out.print(mark + " ");
                }
                System.out.println();
        }
}
```



Main.java

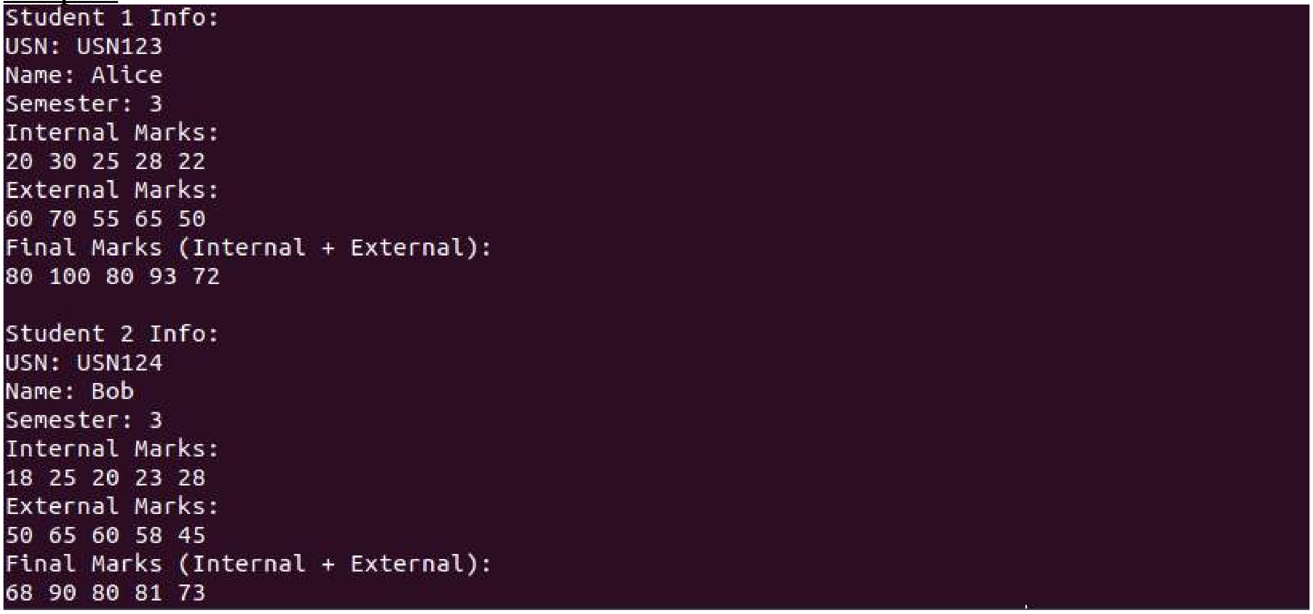
```
import cie.Internals;
import see.External;
public class Main{
    public static void main (String []args){
        int n=2;
        int[] internalMarks1= {20,30,25,28,22};
        int[] externalMarks1= {60,70,55,65,50};
        Internals student1Internal = new Internals ("USN123","Alice",
                                                     3, internalMarks1);
        External student1External = new External ("USN123","Alice",3,
                                                   externalMarks1);

        System.out.println(" Student1 Info:");
        student1Internal.displayStudentInfo();
        student1Internal.displayInternalMarks();
        student1External.displayExternalMarks();
        int[] finalMarks1 = calculateFinalMarks(student1Internal.
                          internalMarks, student1External.
                                         externalMarks);

        displayFinalMarks(finalMarks1);
    }
    public static int[] calculateFinalMarks (int[] internalMarks,
                                             int[] externalMarks){
        int[] finalMarks = new int[5];
        for (int i=0; i<5; i++){
                finalMarks[i]= internalMarks[i]
                             + externalMarks[i];
        }
        return finalMarks;
    }
    public static void ~~mon~~ displayFinalMarks (int[] finalMarks){
        System.out.println("Final Marks (Internal + External):");
        for (int mark : finalMarks){
            System.out.print(mark+ " ");
        System.out.println();
```

O/P
3
3
23/11/23
3

Code:

```
//Student.java
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 displayStudentInfo() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
//Internals.java
package cie;

public class Internals extends Student {
    public int[] internalMarks = new int[5];

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

    public void displayInternalMarks() {
        System.out.println("Internal Marks: ");
        for (int mark : internalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }
}
//Externals.java
package see;

import cie.Student;

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

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

    public void displayExternalMarks() {
        System.out.println("External Marks: ");
        for (int mark : externalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }
}
//Main.java
import cie.Internals;
import see.External;

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

        int n = 2;

        int[] internalMarks1 = {20, 30, 25, 28, 22};
        int[] externalMarks1 = {60, 70, 55, 65, 50};
        Internals student1Internal = new Internals("USN123", "Alice", 3, internalMarks1);
        External student1External = new External("USN123", "Alice", 3, externalMarks1);

        int[] internalMarks2 = {18, 25, 20, 23, 28};
        int[] externalMarks2 = {50, 65, 60, 58, 45};
        Internals student2Internal = new Internals("USN124", "Bob", 3, internalMarks2);
        External student2External = new External("USN124", "Bob", 3, externalMarks2);

        System.out.println("Student 1 Info: ");
        student1Internal.displayStudentInfo();
        student1Internal.displayInternalMarks();
        student1External.displayExternalMarks();

        int[] finalMarks1 = calculateFinalMarks(student1Internal.internalMarks,
student1External.externalMarks);
        displayFinalMarks(finalMarks1);

        System.out.println("\nStudent 2 Info: ");
        student2Internal.displayStudentInfo();
        student2Internal.displayInternalMarks();
        student2External.displayExternalMarks();
        int[] finalMarks2 = calculateFinalMarks(student2Internal.internalMarks,
student2External.externalMarks);
        displayFinalMarks(finalMarks2);
    }

    public static int[] calculateFinalMarks(int[] internalMarks, int[] externalMarks) {
        int[] finalMarks = new int[5];
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = internalMarks[i] + externalMarks[i];
        }
        return finalMarks;
    }


    public static void displayFinalMarks(int[] finalMarks) {
        System.out.println("Final Marks (Internal + External): ");
        for (int mark : finalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }
}
```

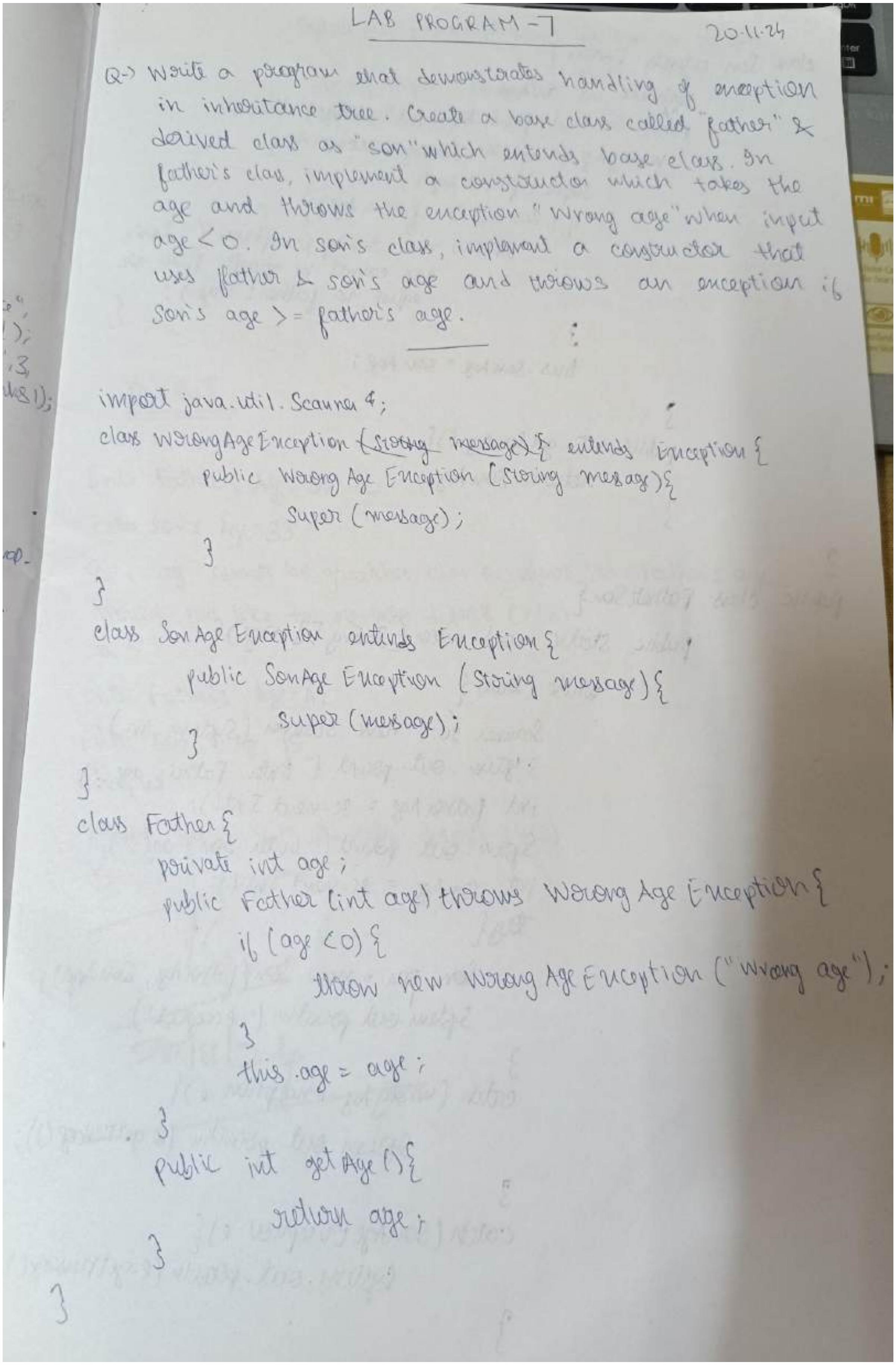
Output:



```
Student 1 Info:
USN: USN123
Name: Alice
Semester: 3
Internal Marks:
20 30 25 28 22
External Marks:
60 70 55 65 50
Final Marks (Internal + External):
80 100 80 93 72

Student 2 Info:
USN: USN124
Name: Bob
Semester: 3
Internal Marks:
18 25 20 23 28
External Marks:
50 65 60 58 45
Final Marks (Internal + External):
68 90 80 81 73
```

Program-7
Exceptions

Observation:

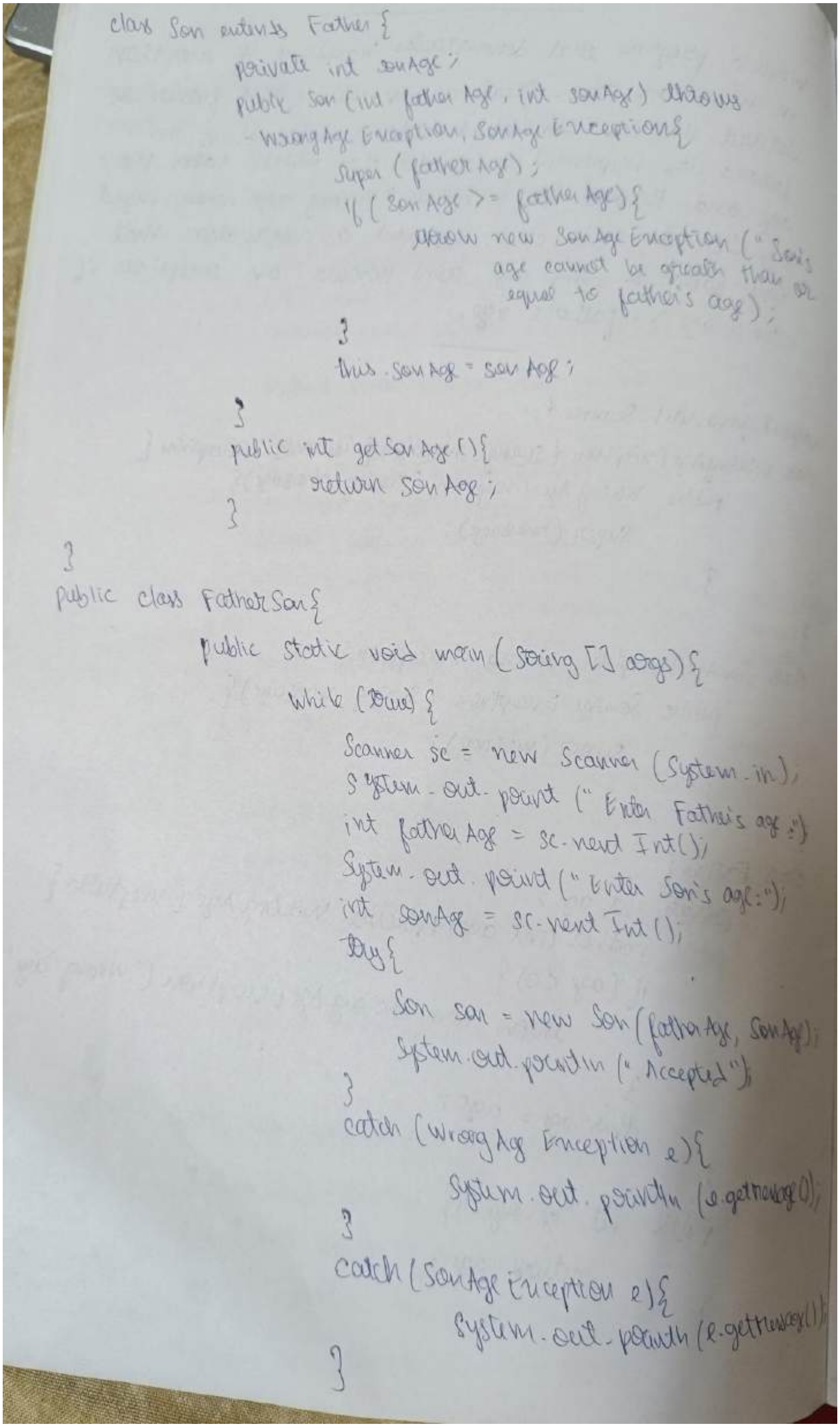


LAB PROGRAM - 7

20-11-24

Q-> Write a program that demonstrates handling of exception in inheritance tree. Create a base class called "father" & derived class as "son" which extends base class. In father's class, implement a constructor which takes the age and throws the exception "Wrong age" when input age < 0. In son's class, implement a constructor that uses father & son's age and throws an exception if son's age >= father's age.

```
import java.util.Scanner *;
class WrongAgeException ~~(String message){~~ extends Exception {
        public WrongAgeException (String message){
                Super (message);
        }
}
class SonAgeException extends Exception{
        public SonAgeException (String message){
                super (message);
        }
}
class Father{
        private int age ;
        public Father (int age) throws WrongAgeException{
                if (age <0) {
                        throw new WrongAgeException ("Wrong age");
                }
                this.age = age ;
        }
        public int getAge (){
                return age ;
        }
}
```



```
class Son extends Father {
        private int sonAge;
        public Son (int fatherAge, int sonAge) throws
         WrongAge Exception, SonAge Exception{
                Super (father Age);
                if ( Son Age >= father Age){
                        throw new SonAge Exception (" Son's
                                age cannot be greater than or
                                equal to father's age);
                }
                this.sonAge = son Age;
        }
        public int getSonAge(){
                return Son Age;
        }
}
public class FatherSon{
        public static void main (String [] args){
                while (true) {
                        Scanner sc = new Scanner (System.in);
                        System.out.print (" Enter Father's age :");
                        int father Age = sc.next Int();
                        System.out.print (" Enter Son's age:");
                        int sonAge = sc.next Int ();
                        try{
                                Son son = new Son (fatherAge, SonAge);
                                System.out.println (" Accepted");
                        }
                        catch (WrongAge Exception e){
                                System.out.println (e.getMessage());
                        }
                        catch (SonAge Exception e){
                                System.out.println (e.getMessage());
                        }
```

```
            System.out.println(" Would you like to re-enter
                                             details (Y/n)");
            String input = sc.next();
            if(input.equalsIgnorCase("n")){
                break;
            }
        }
    }
}
```

OUTPUT

```
Enter Father's Age: 88 22
Enter Son's Age: 33
Son's age cannot be greather than or equal to Father's age
Would you like to re-enter details (Y/n)
Y
Enter Father's Age: 42
Enter Son's Age: 15
Accepted
Would you like to re-enter details (Y/n)
n
```

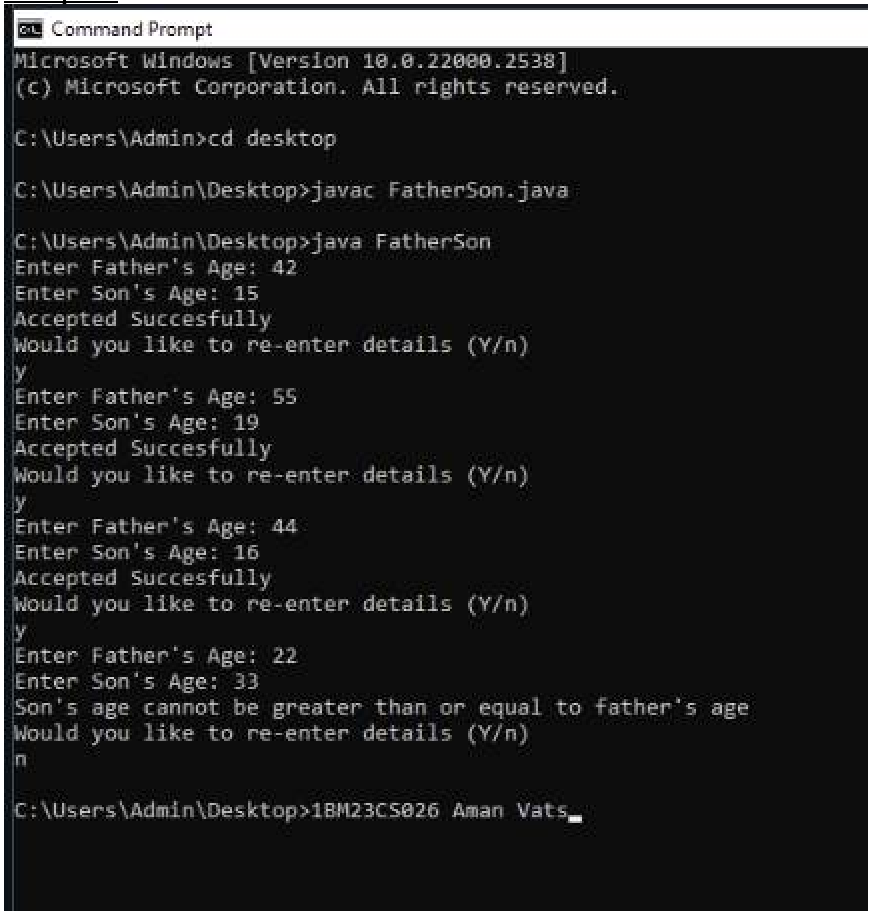
20/11/24

Code:

```
import java.util.Scanner;
class WrongAgeException extends Exception {
   public WrongAgeException(String message) {
      super(message);
   }
}

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

class Father {
   private int age;
   public Father(int age) throws WrongAgeException {
      if (age < 0) {
         throw new WrongAgeException("Wrong age");
      }
      this.age = age;
   }
   public int getAge() {
      return age;
   }
}

class Son extends Father {
   private int sonAge;
   public Son(int fatherAge, int sonAge) throws WrongAgeException, SonAgeException {
      super(fatherAge);
      if (sonAge >= fatherAge) {
         throw new SonAgeException("Son's age cannot be greater than or equal to father's age");
      }
      this.sonAge = sonAge;
   }
   public int getSonAge() {
      return sonAge;
   }
}
public class FatherSon{
   public static void main(String[] args) {
      while(true){
         Scanner sc = new Scanner(System.in);
         System.out.print("Enter Father's Age: ");
         int fatherAge = sc.nextInt();
```

```
            System.out.print("Enter Son's Age: ");
            int sonAge = sc.nextInt();
            try {
                Son son = new Son(fatherAge, sonAge);
                System.out.println("Accepted Succesfully");
            }
            catch (WrongAgeException e) {
                System.out.println(e.getMessage());
            }
            catch (SonAgeException e) {
                System.out.println(e.getMessage());
            }
            System.out.println("Would you like to re-enter details (Y/n)");
            String input = sc.next();
            if (input.equalsIgnoreCase("n")) {
                break;
            }
        }
    }
}
```

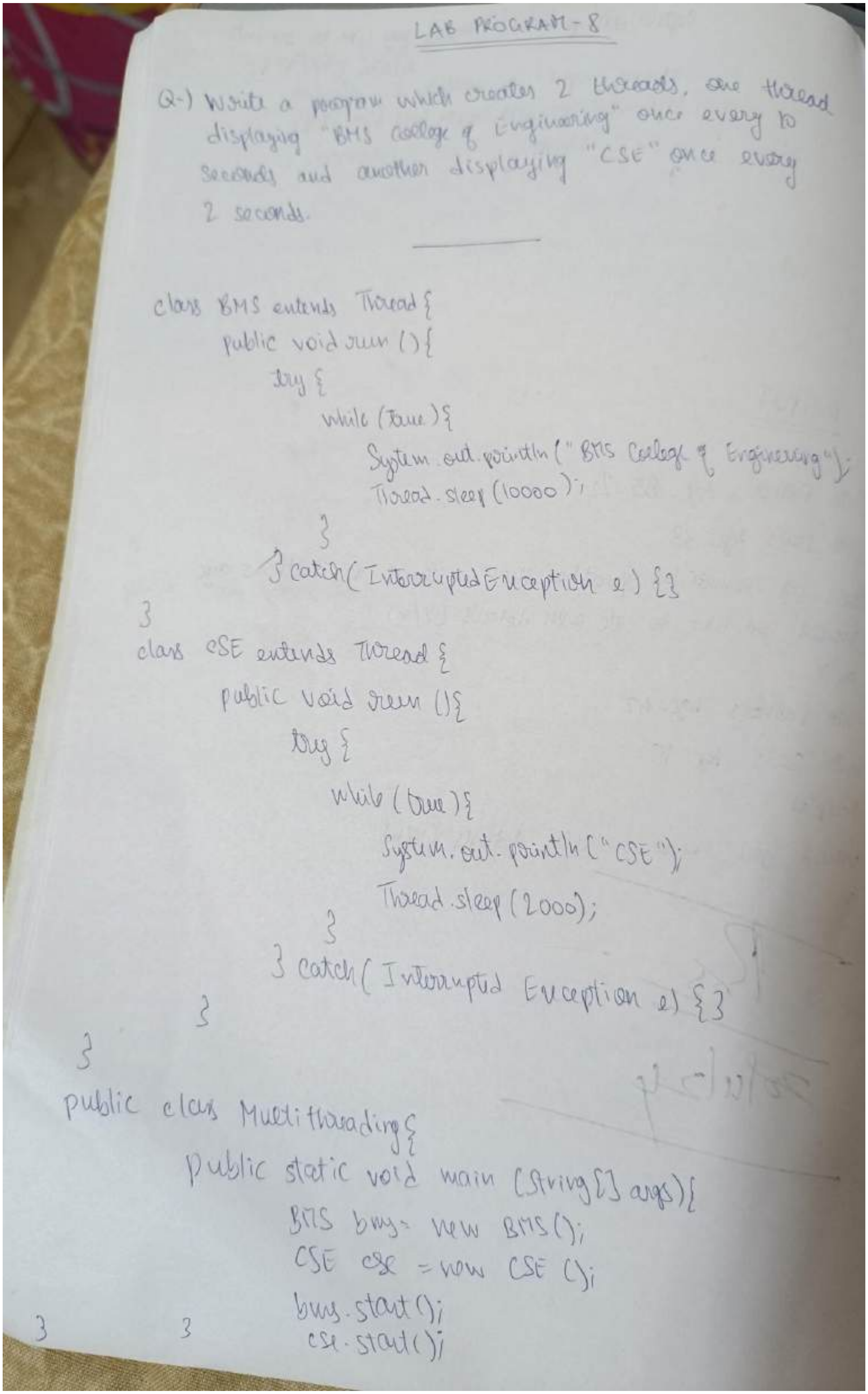
Output:



```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac FatherSon.java

C:\Users\Admin\Desktop>java FatherSon
Enter Father's Age: 42
Enter Son's Age: 15
Accepted Succesfully
Would you like to re-enter details (Y/n)
y
Enter Father's Age: 55
Enter Son's Age: 19
Accepted Succesfully
Would you like to re-enter details (Y/n)
y
Enter Father's Age: 44
Enter Son's Age: 16
Accepted Succesfully
Would you like to re-enter details (Y/n)
y
Enter Father's Age: 22
Enter Son's Age: 33
Son's age cannot be greater than or equal to father's age
Would you like to re-enter details (Y/n)
n

C:\Users\Admin\Desktop>1BM23CS026 Aman Vats_
```

Program-8
Threads

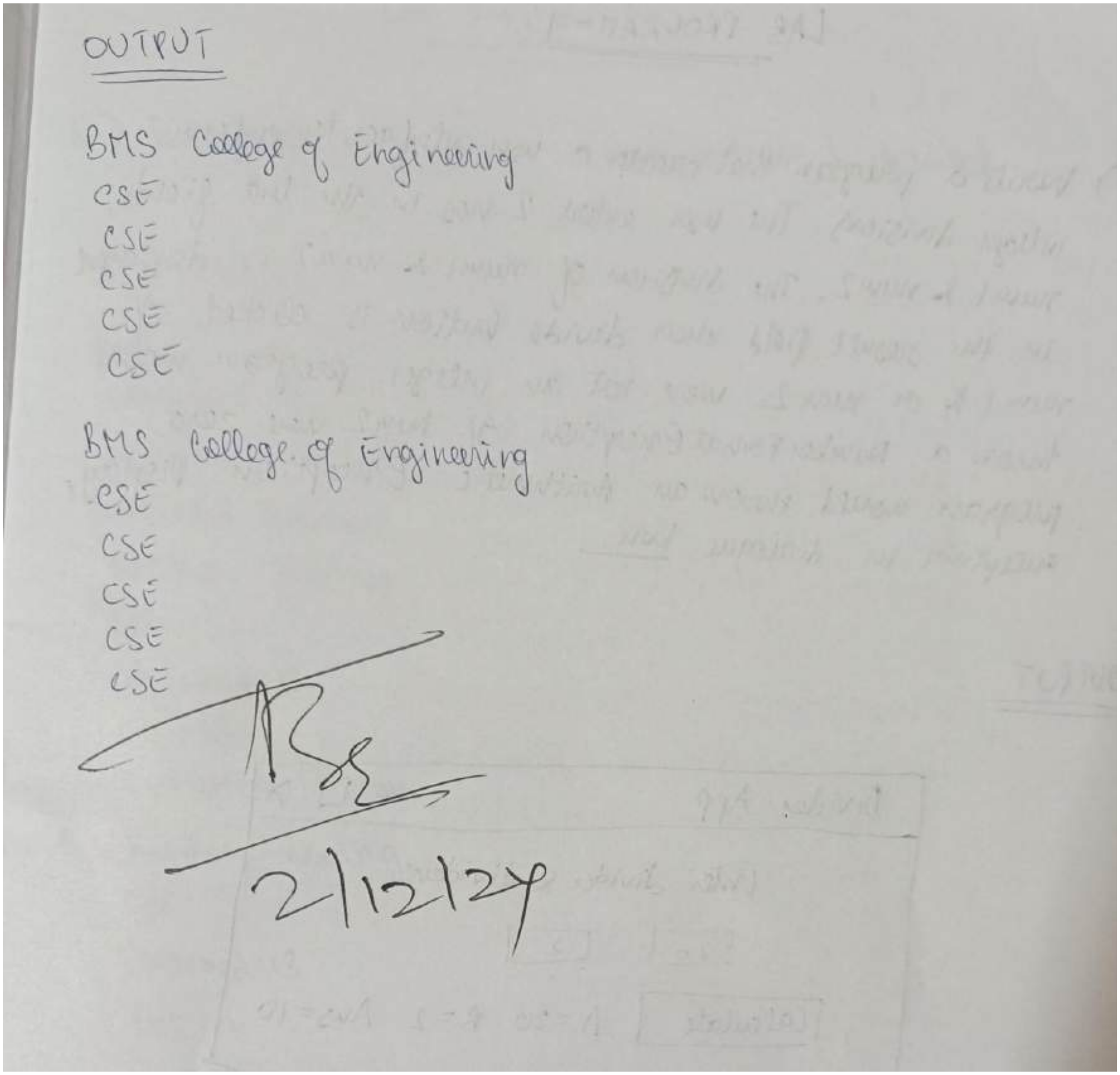
Observation:



LAB PROGRAM-8

Q-) Write a program which creates 2 threads, one thread displaying "BMS college of Engineering" once every 10 seconds and another displaying "CSE" once every 2 seconds.

```
class BMS extends Thread{
    public void run (){
        try {
            while (true){
                System.out.println("BMS College of Engineering");
                Thread.sleep (10000);
            }
        } catch(InterruptedException e) {}
}
class CSE extends Thread {
    public void run (){
        try {
            while (true){
                System.out.println ("CSE");
                Thread.sleep (2000);
            }
        } catch( Interrupted Exception e) {}
    }
}
public class Multithreading{
    public static void main (String[] args){
        BMS bms= new BMS();
        CSE cse = new CSE ();
        bms.start();
}   }   cse.start();
```



OUTPUT

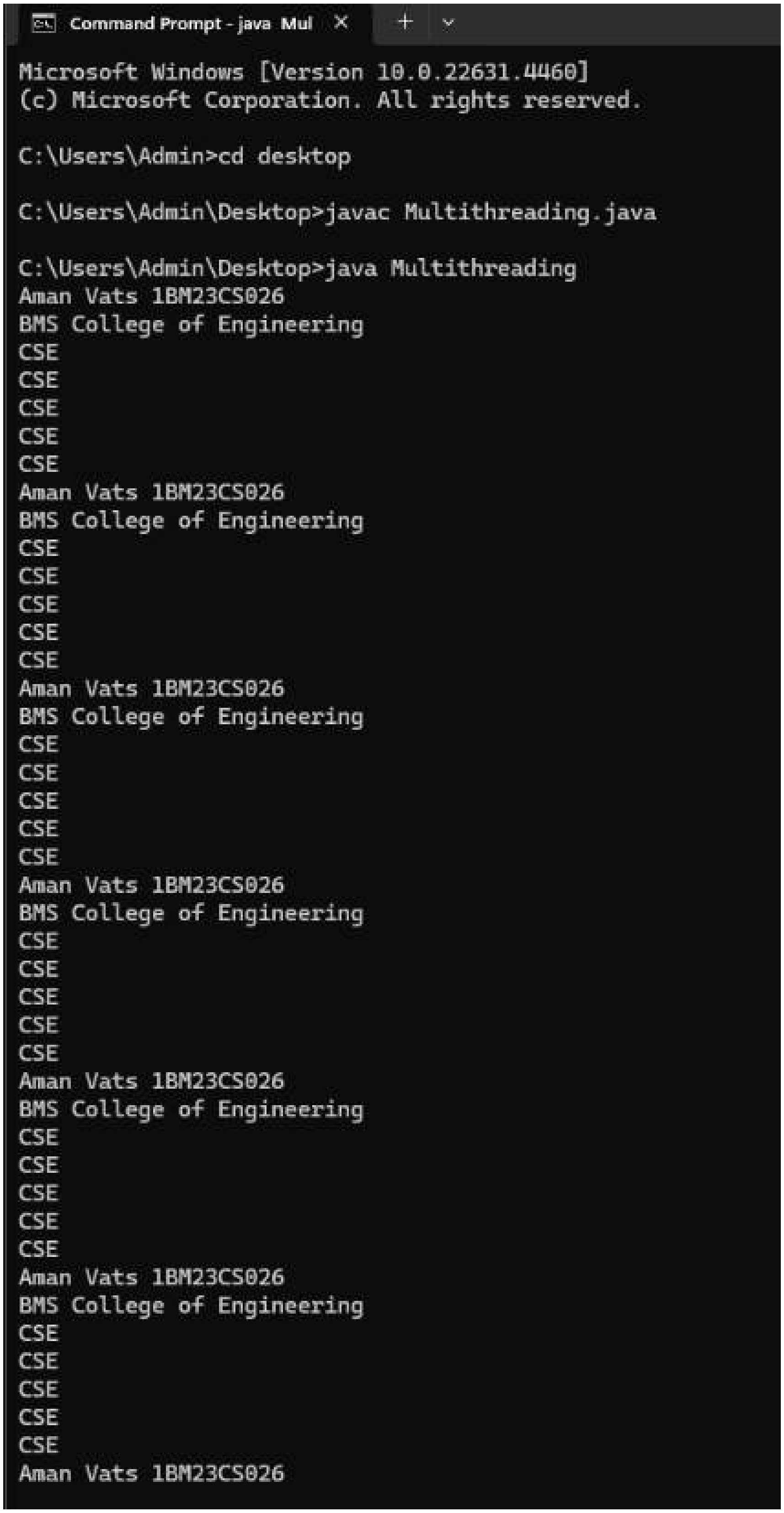
```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```

2/12/24

Code:

```
class BMS extends Thread {
    public void run() {
        try {
            while (true) {
                        System.out.println("Aman Vats 1BM23CS026");
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000); // Sleep for 10 seconds
            }
        }catch (InterruptedException e) {}
    }
}
class CSE extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000); // Sleep for 2 seconds
            }
        }catch (InterruptedException e) {}
    }
}

public class Multithreading{
    public static void main(String[] args) {
        BMS bms = new BMS();
        CSE cse = new CSE();
        bms.start();
        cse.start();
    }
}
```

Output:

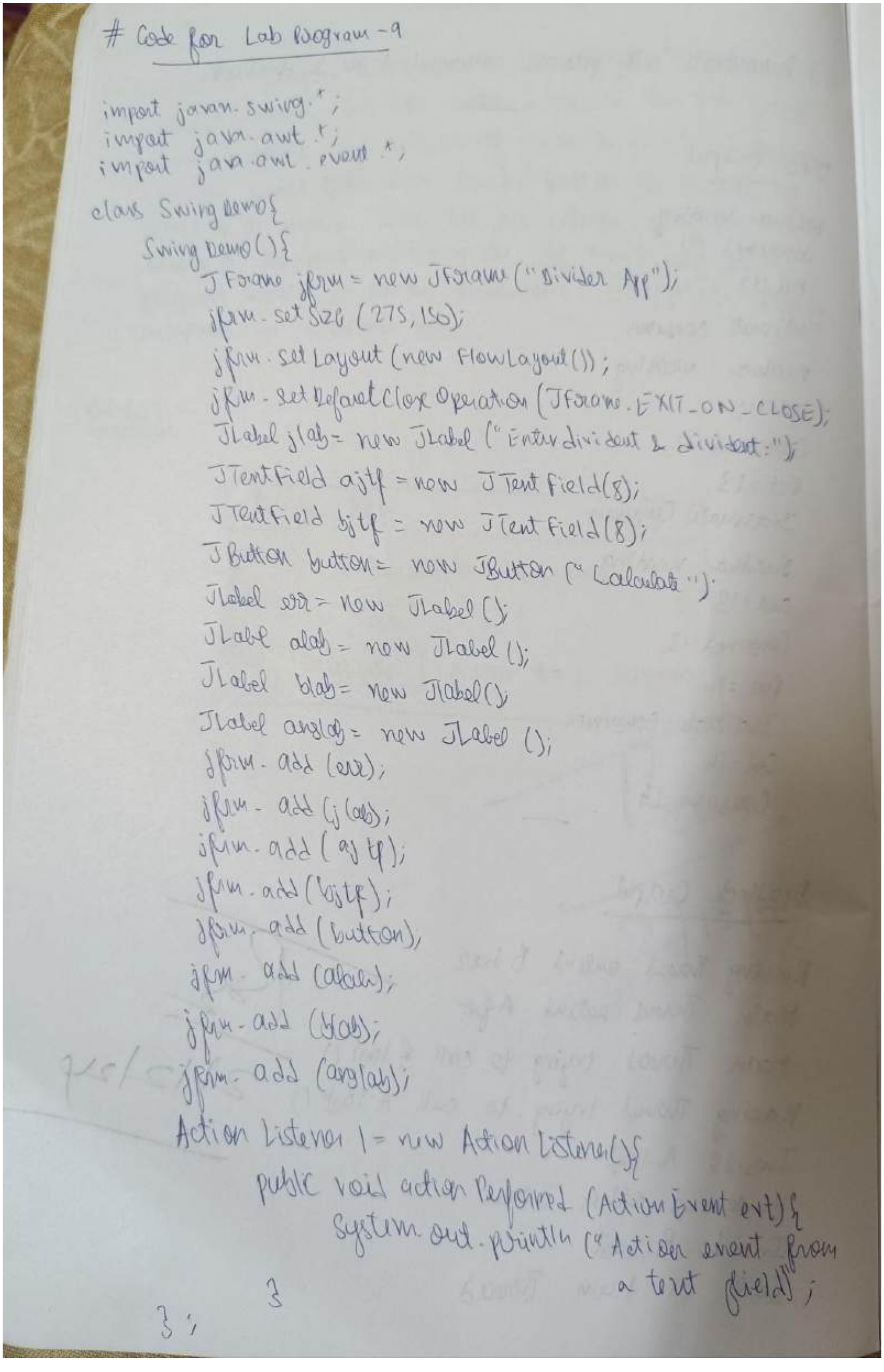


```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac Multithreading.java

C:\Users\Admin\Desktop>java Multithreading
Aman Vats 1BM23CS026
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
Aman Vats 1BM23CS026
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
Aman Vats 1BM23CS026
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
Aman Vats 1BM23CS026
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
Aman Vats 1BM23CS026
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
Aman Vats 1BM23CS026
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
Aman Vats 1BM23CS026
```

Program-9

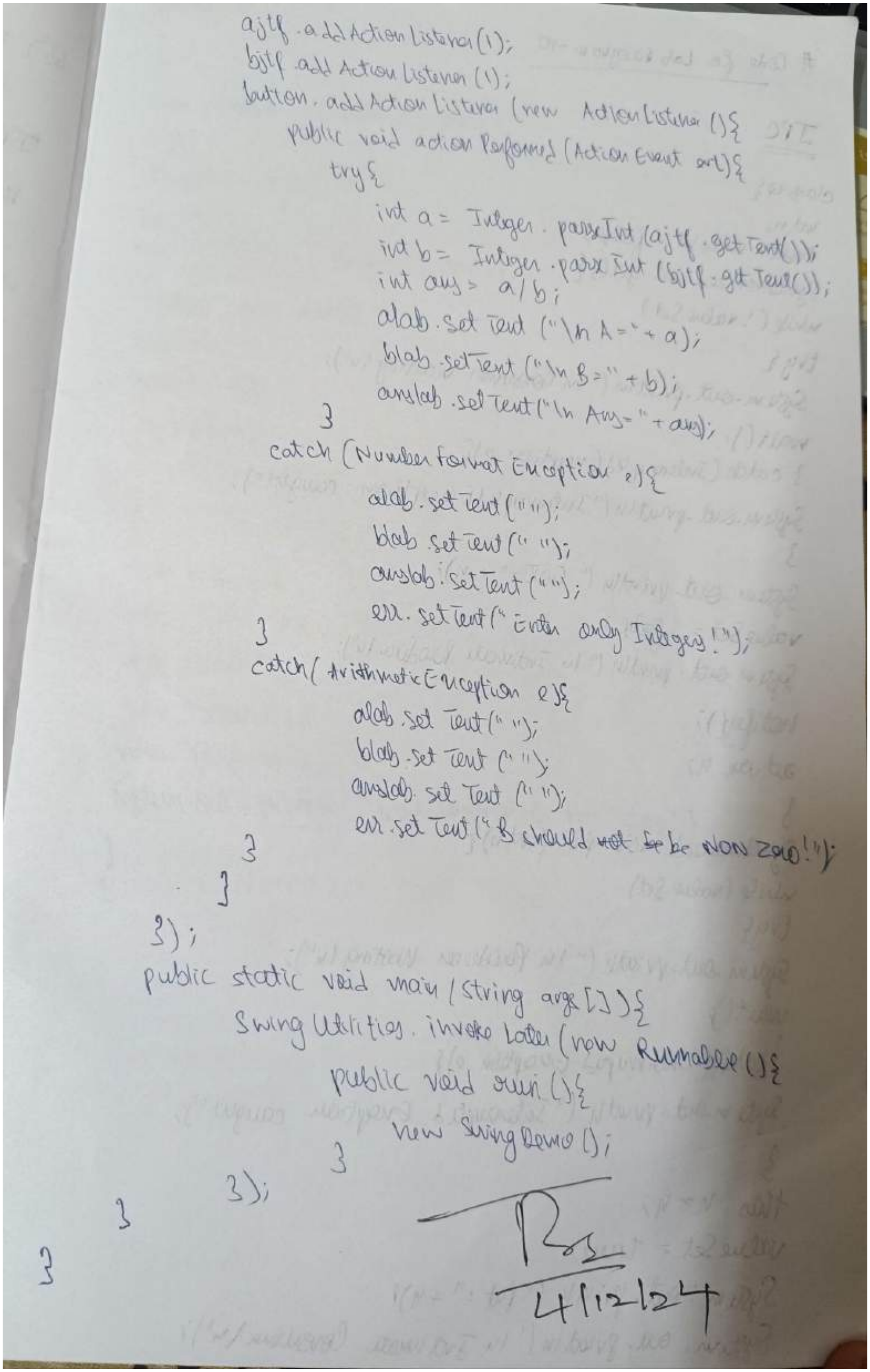
User interface to perform integer division

Observation:



Code for Lab Program - 9

```
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 divident & divisor:");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        JButton button= new JButton("Calculate").
        JLabel err= new JLabel();
        JLable alab= new JLabel();
        JLabel blab= new JLabel();
        JLabel anslab= new JLabel ();
        jfrm.add (err);
        jfrm.add (jlab);
        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");
            }
    };
```



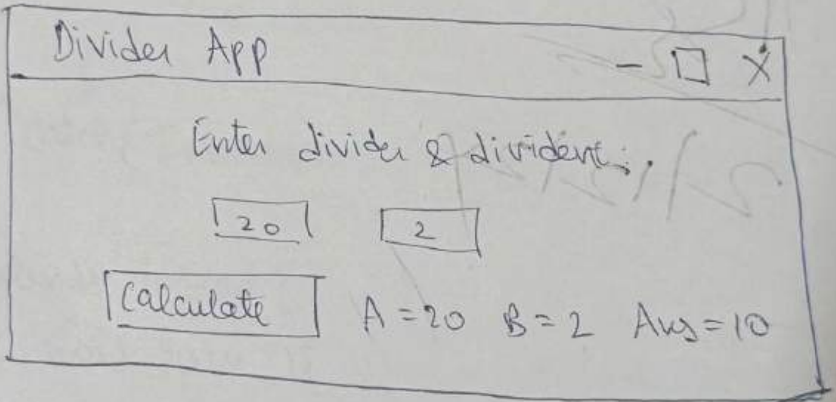
```
ajtf.addActionListener(1);
bjtf.addActionListener(1);
button.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent evt){
        try{
            int a= Integer.parseInt(ajtf.getText());
            int b= Integer.parseInt(bjtf.getText());
            int ans= a/b;
            alab.setText("\n A="+a);
            blab.setText("\n B="+b);
            anslab.setText("\n Ans="+ans);
        }
    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 ~~not~~ ~~to~~ be Non Zero!");
    }
  }
});
public static void main(String args[]){
    SwingUtilities.invokeLater(new Runnable(){
        public void run(){
            new SwingDemo();
        }
    });
}
}
```

4/12/24

LAB PROGRAM-9

Q-) Write a program that creates a user interfac to perform integer divisions. The user enters 2 nos. in the text fields, num1 & num2. The division of num1 & num2 is displayed in the result field when divide button is clicked. If num1 & or num2 were not an integer, program would throw a NumberFormatException. If Num2 were zero, program would throw an Arithmetic Exception Display exception in dialogue box.

OUTPUT



Enter divider & divident :

20 2

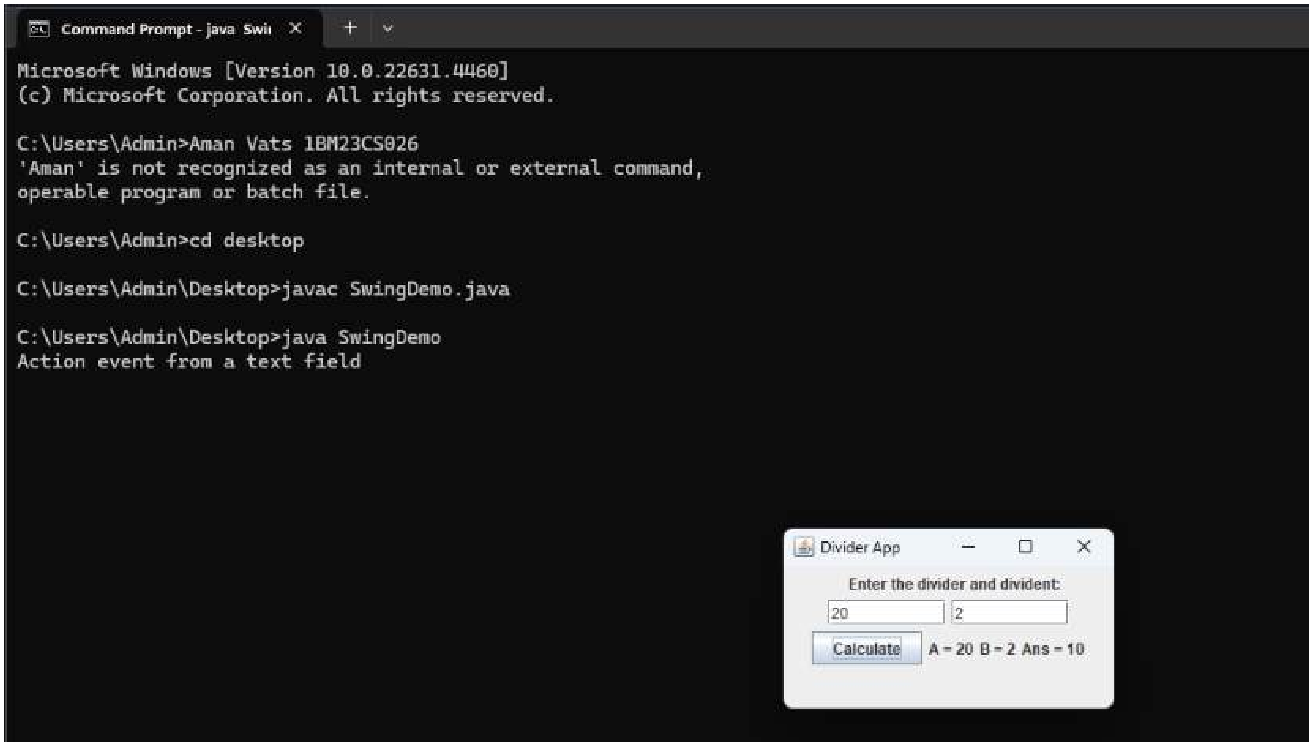
A = 20 B = 2 Ans = 10

2/12/24

Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
SwingDemo(){
// create jframe container
JFrame jfrm = new JFrame("Divider App");
jfrm.setSize(275, 150);
jfrm.setLayout(new FlowLayout());
// to terminate on close
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// text label
JLabel jlab = new JLabel("Enter the divider and divident:");
// add text field for both numbers
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);
// calc button
JButton button = new JButton("Calculate");
// labels
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();

JLabel anslab = new JLabel();
// add in order :)
jfrm.add(err); // to display error bois
jfrm.add(jlab);
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("\nA = " + a);
blab.setText("\nB = " + b);
anslab.setText("\nAns = "+ ans);
}
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!");
}
}
});
// display frame
jfrm.setVisible(true);
}
public static void main(String args[]){
// create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable(){
public void run(){
new SwingDemo();
}
});
}
}
```

Output:



```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>Aman Vats 1BM23CS026
'Aman' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac SwingDemo.java

C:\Users\Admin\Desktop>java SwingDemo
Action event from a text field
```

Enter the divider and divident:

20

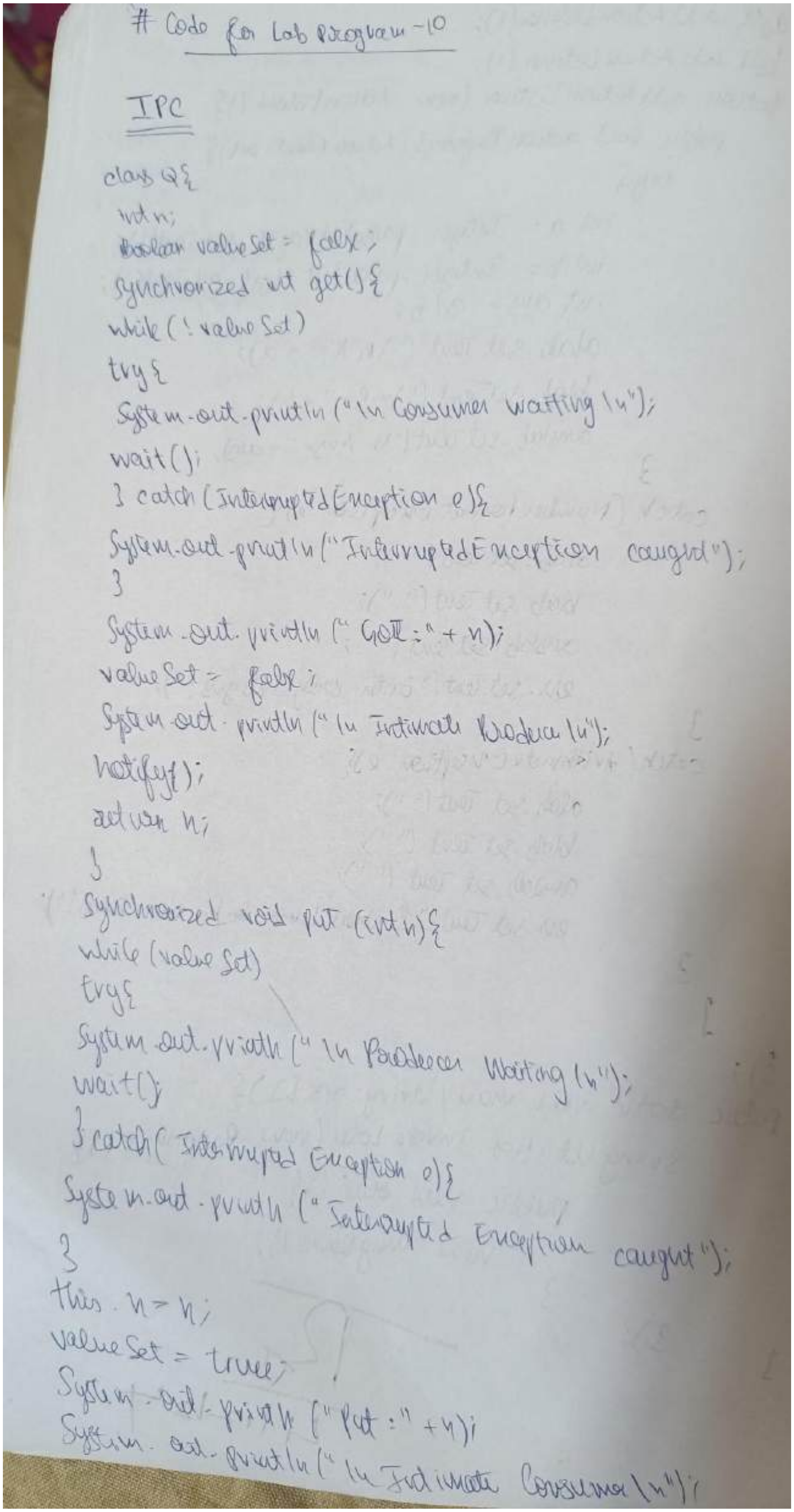
2

A = 20 B = 2 Ans = 10

Program-10

IPC, DEADLOCK

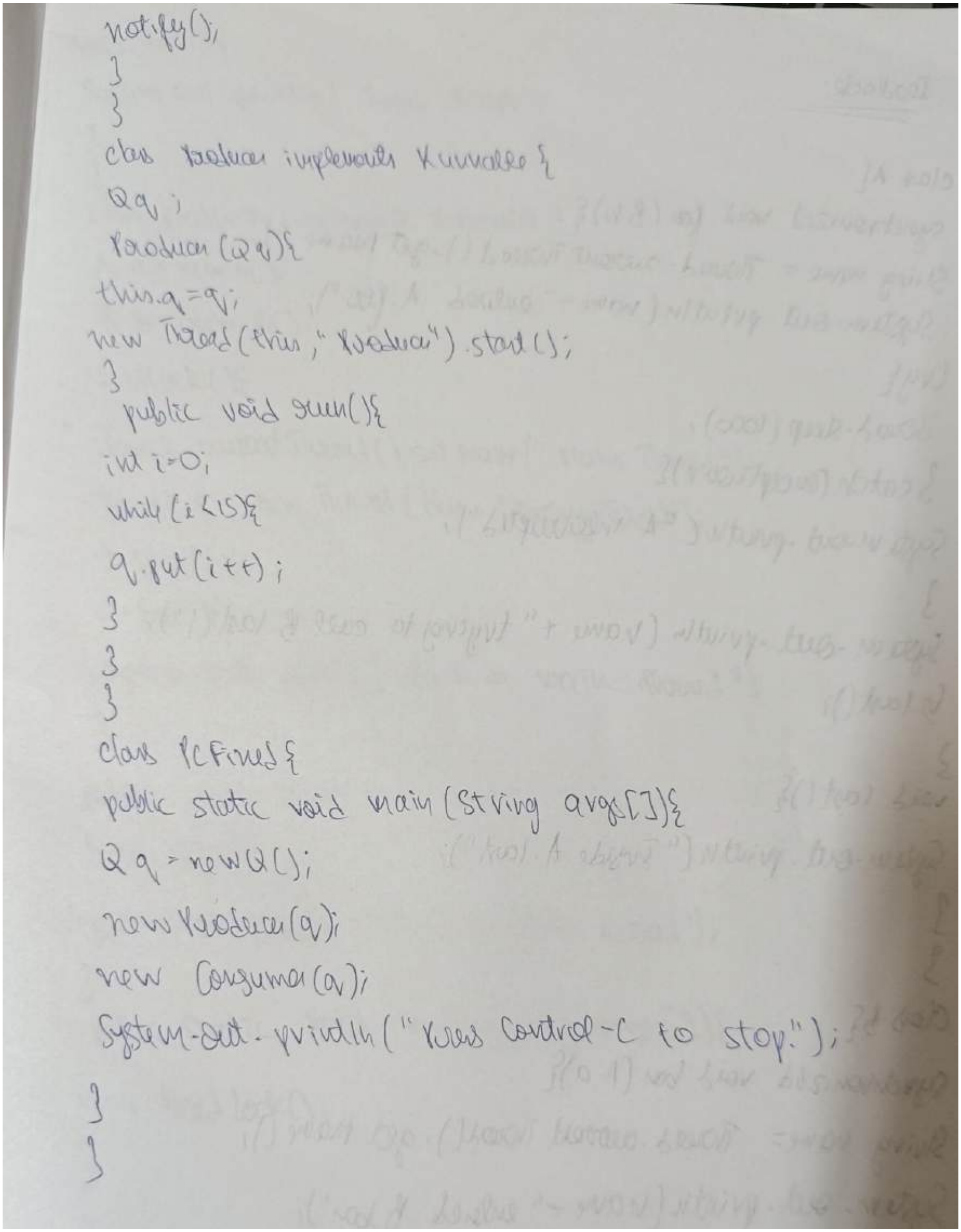
Observation:



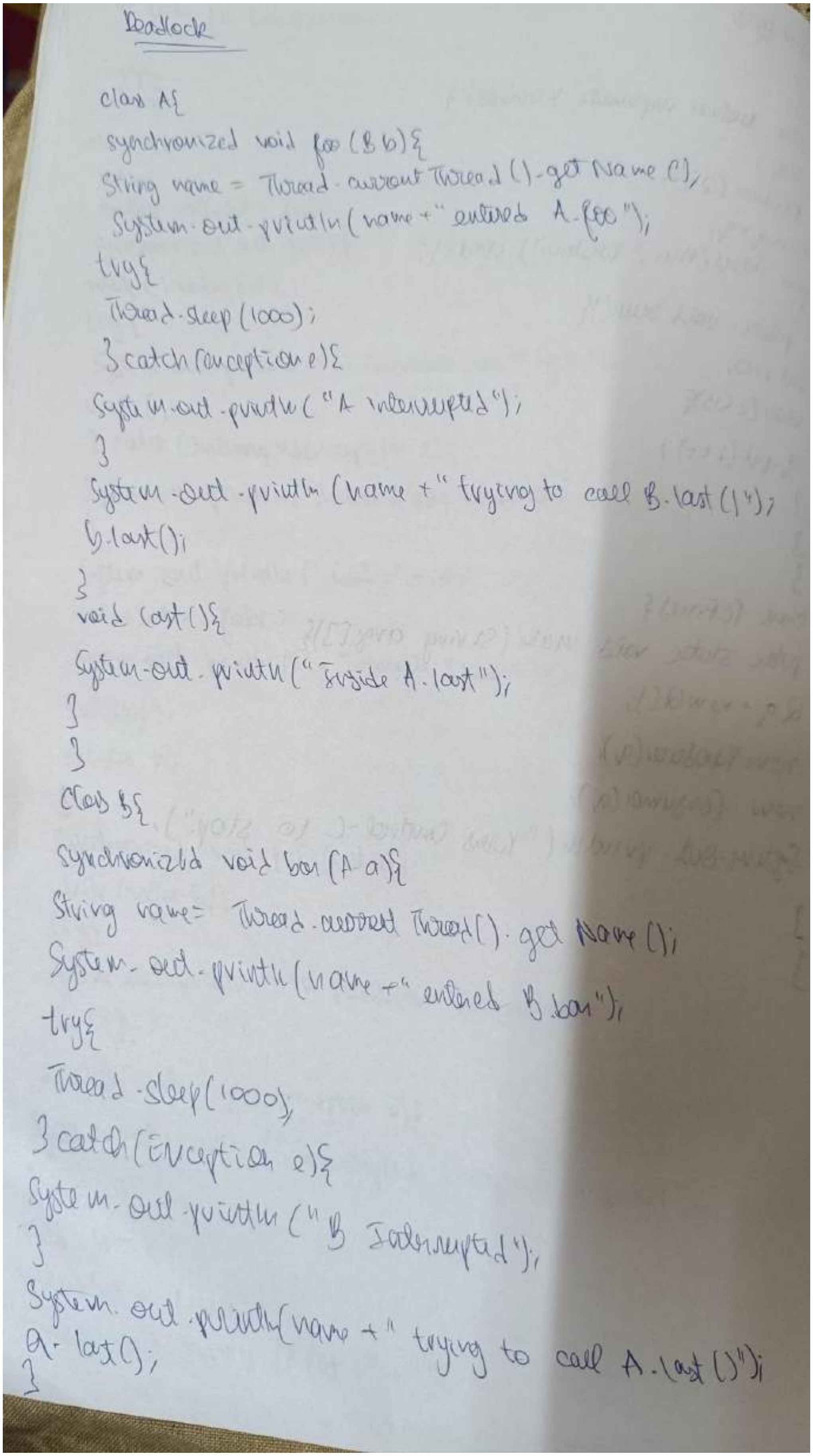
Code for Lab Program -10

IPC

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

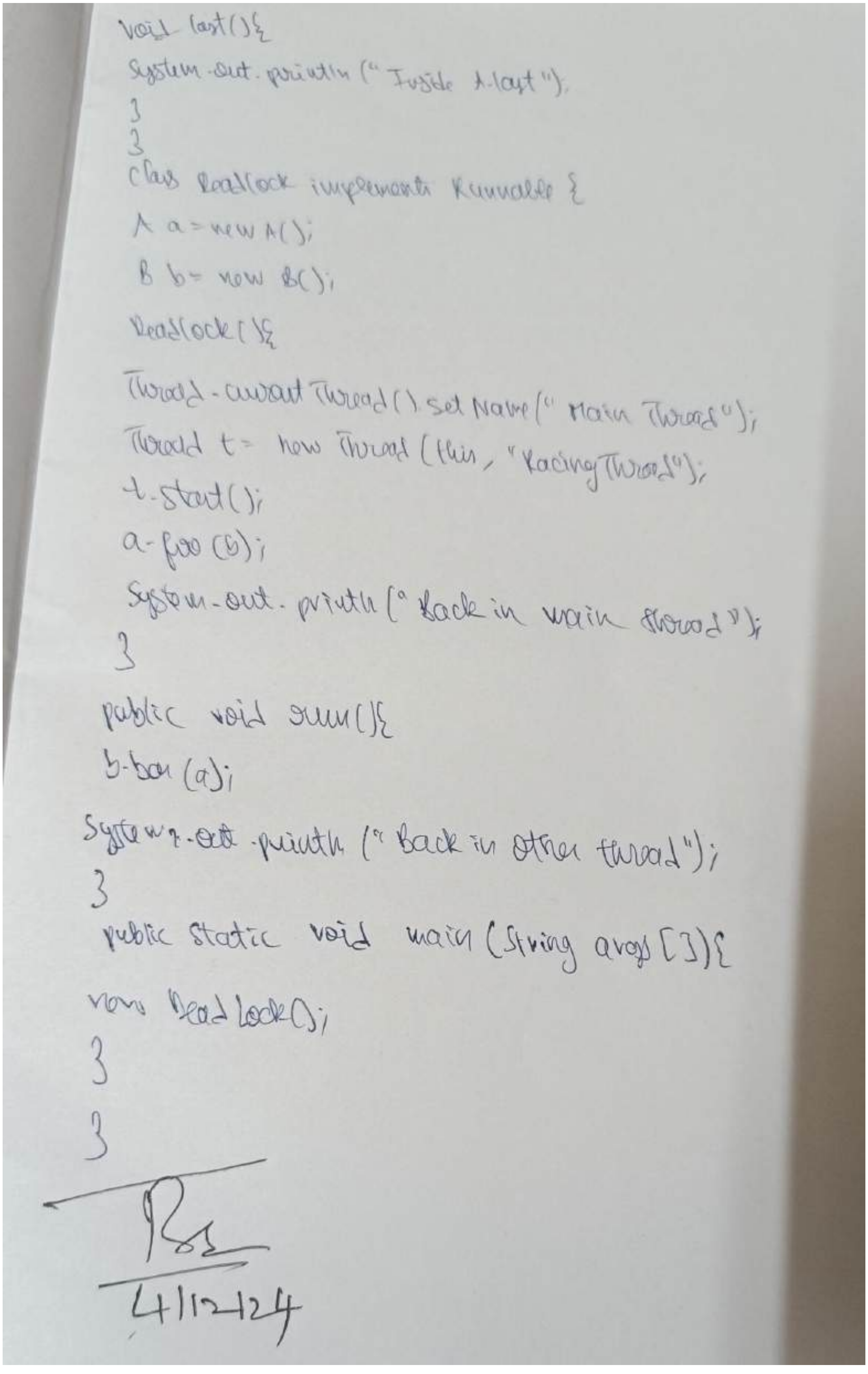


```
notify();
}
}
class Producer implements Runnable{
Q q;
Producer(Q q){
this.q=q;
new Thread(this,"Producer").start();
}
public void run(){
int i=0;
while(i<15){
q.put(i++);
}
}
}
class PCFixed{
public static void main(String args[]){
Q q=new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press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



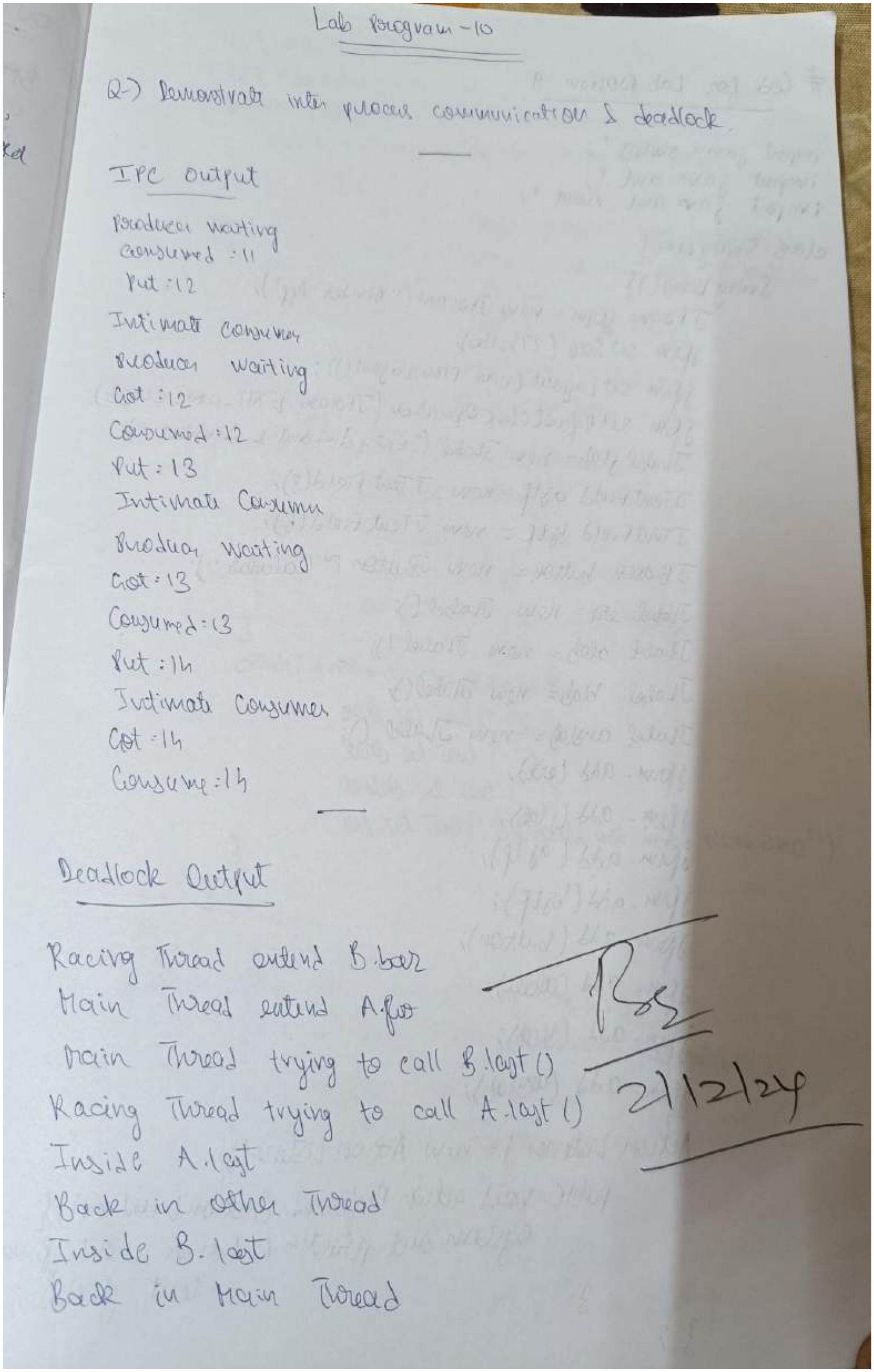
Deadlock

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



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

4/12/24



Lab Program-10

Q-) Demonstrate inter process communication & deadlock.

IPC output

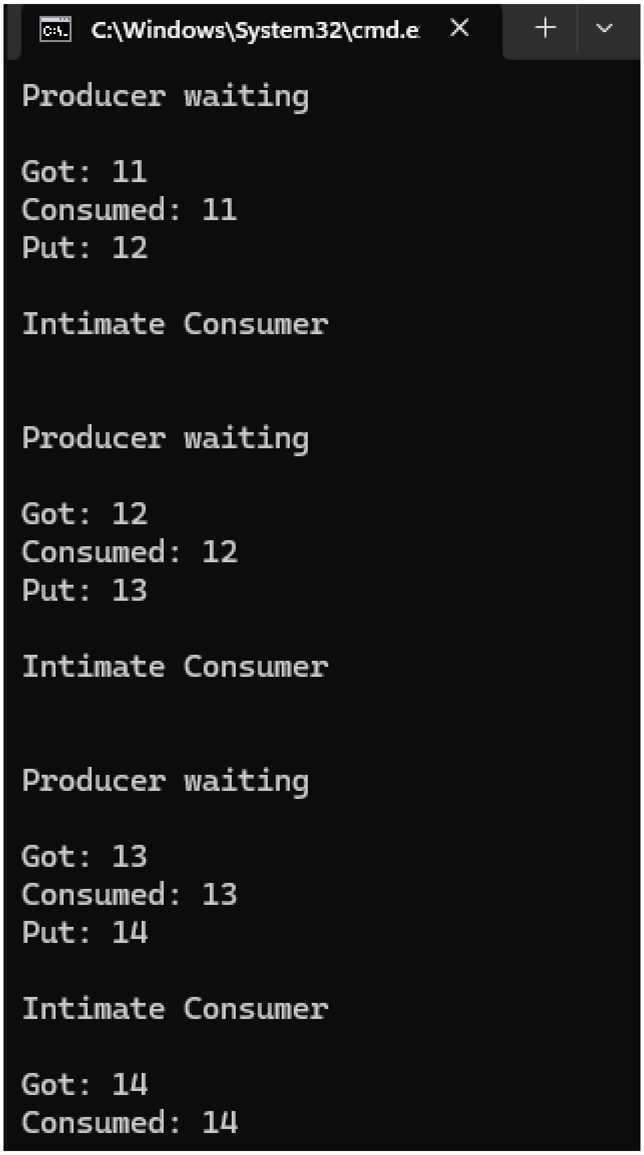
```
Producer waiting
Consumed : 11
Put : 12
Intimate consumer
Producer waiting
Got : 12
Consumed : 12
Put : 13
Intimate Consumer
Producer waiting
Got : 13
Consumed : 13
Put : 14
Intimate Consumer
Got : 14
Consume : 14
```

Deadlock Output

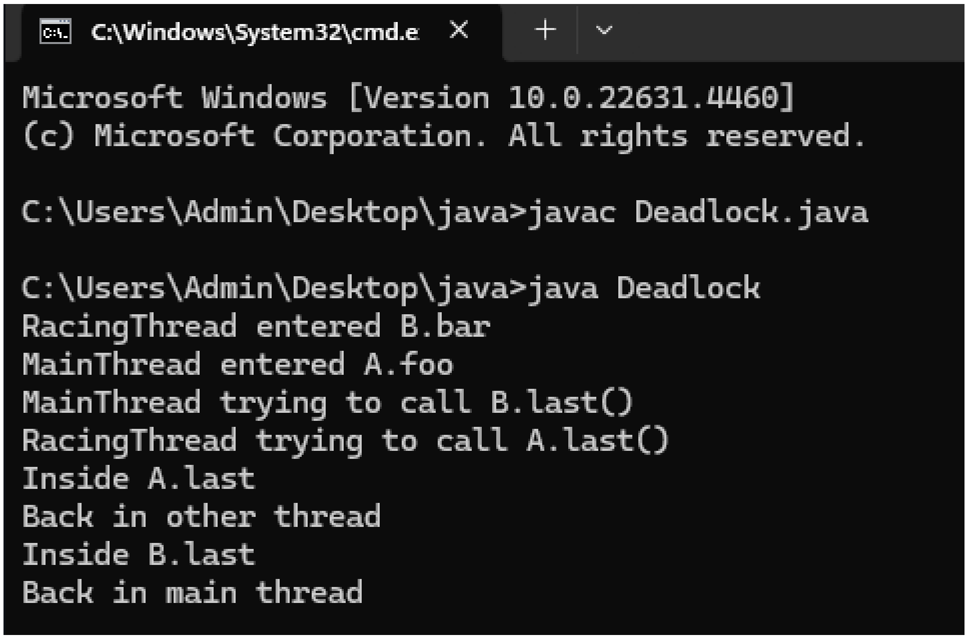
```
Racing Thread entered B.bar
Main Thread entered A.foo
main Thread trying to call B.last()
Racing Thread trying to call A.last()
Inside A.last
Back in other Thread
Inside B.last
Back in Main Thread
```

2/12/24

Output:



```
Producer waiting

Got: 11
Consumed: 11
Put: 12

Intimate Consumer


Producer waiting

Got: 12
Consumed: 12
Put: 13

Intimate Consumer


Producer waiting

Got: 13
Consumed: 13
Put: 14

Intimate Consumer

Got: 14
Consumed: 14
```



```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\java>javac Deadlock.java

C:\Users\Admin\Desktop\java>java Deadlock
RacingThread entered B.bar
MainThread entered A.foo
MainThread trying to call B.last()
RacingThread trying to call A.last()
Inside A.last
Back in other thread
Inside B.last
Back in main thread
```