VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Aamal Mohsin Magdum (1BM23CS002)

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 **Aamal Mohsin Magdum (1BM23CS002),** 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.

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

Index

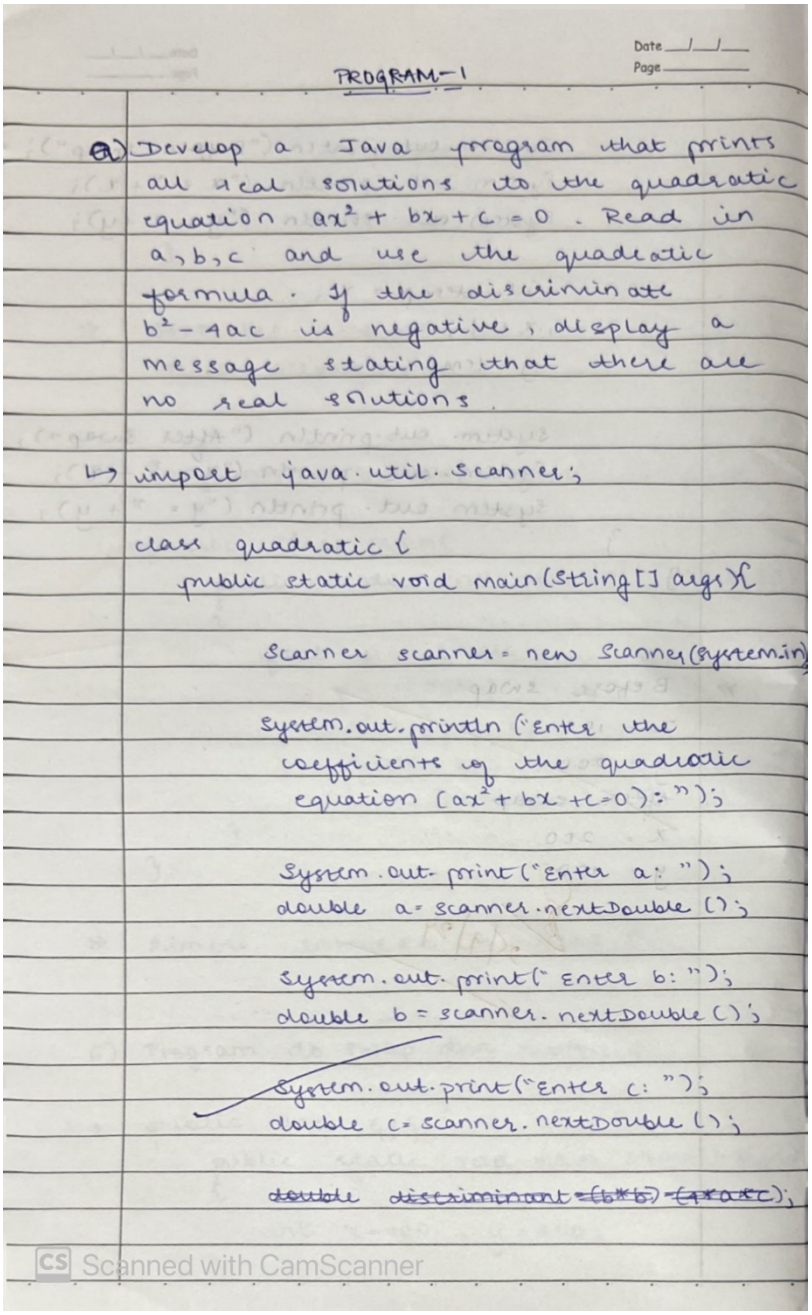
Sl. No.	Date	Experiment Title	Page No.
1	9/10/24	Quadratic equation	5
2	16/10/24	Student class (SGPA)	10
3	23/10/24	Book class (constructors and methods)	17
4	23/10/24	Shape class (abstracts)	23
5	30/10/24	Bank class (inheritance)	30
6	13/11/24	Student marks (package)	41
7	20/11/24	Father-Son age (exception handling)	51
8	27/11/24	Multi-threading	57
9	27/11/24	Deadlock- IPC	60
10	27/11/24	Calculator	64

Github Link:
https://github.com/amalmagdum7/java

Program 1

Develop a java program that prints all real solutions to the quadratic equation ax^2+bx+c=0. Read in a,b,c and use the quadratic formula. If the discriminate b^2- 4ac is negative, display a message stating that there are no real solutions.

Algorithm:



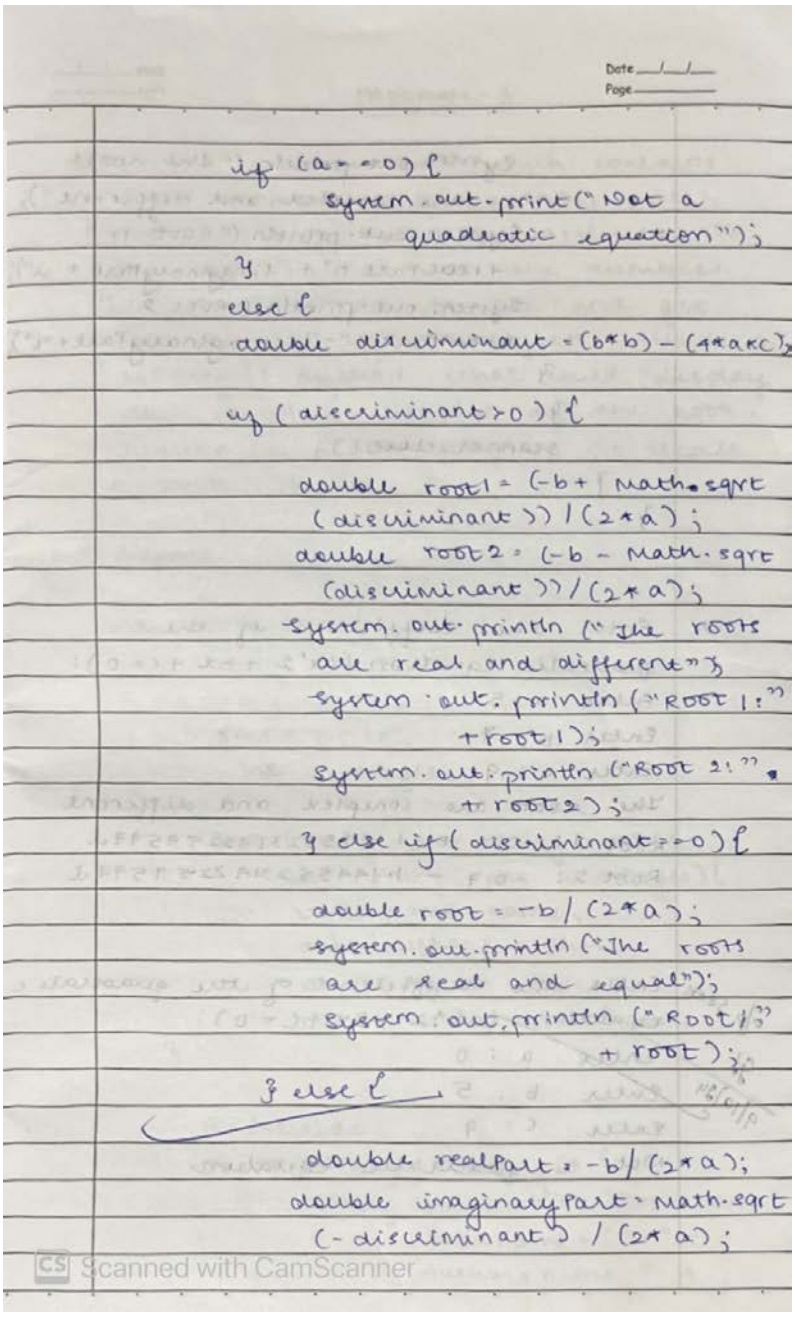
PROGRAM-1

Date __/__/__
Page _____

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

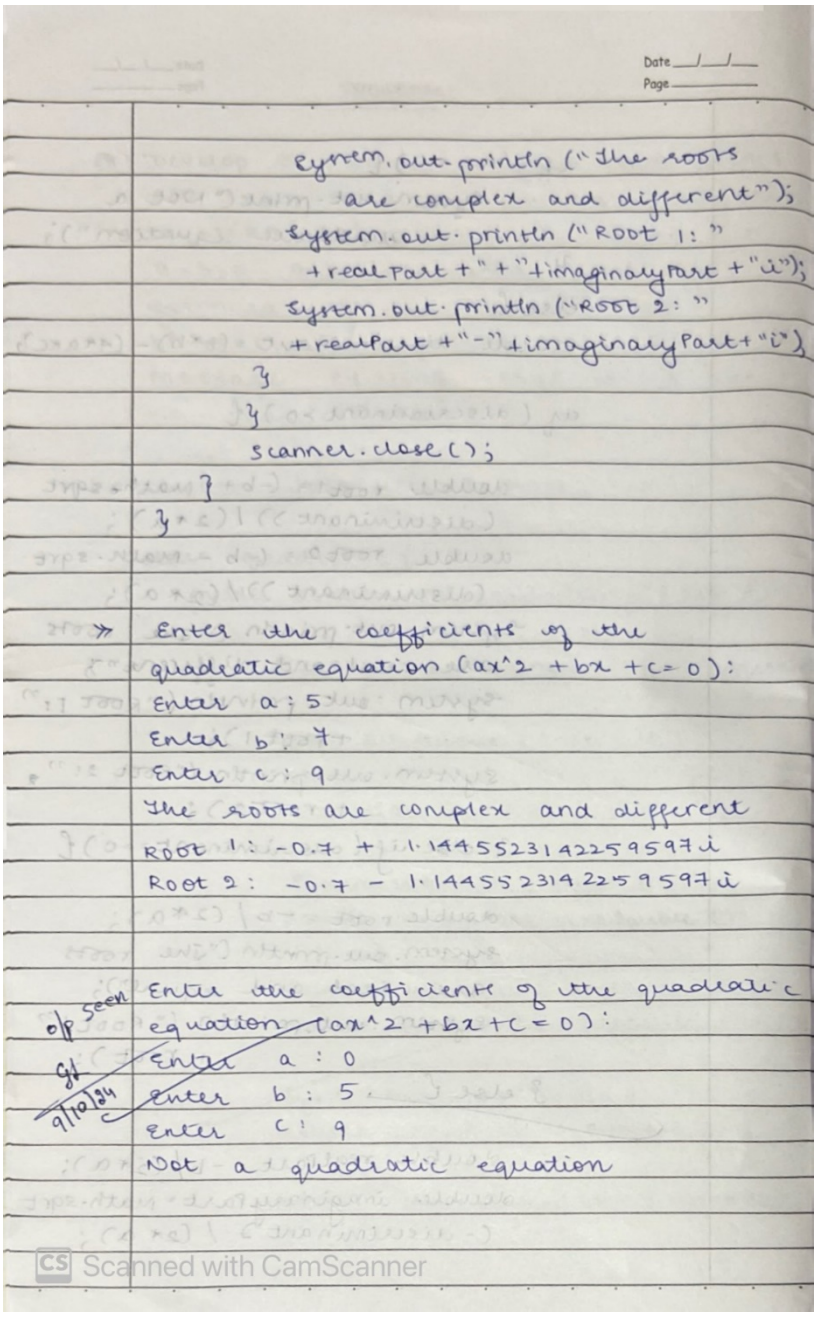
```
import java.util.Scanner;
class quadratic {
    public static void main(String[] args){

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the
        coefficients of the quadratic
        equation (ax^2 + bx + c=0):");

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

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

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

        double discriminant = (b*b) - (4*a*c);
```

CS Scanned with CamScanner



```
if (a == 0) {
    System.out.print("Not a
        quadratic equation");
}
else {
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 equal");
    System.out.println("ROOT:"
            + root);
} else {

    double realPart = -b / (2*a);
    double imaginaryPart = Math.sqrt
    (-discriminant) / (2*a);
```

Scanned with CamScanner



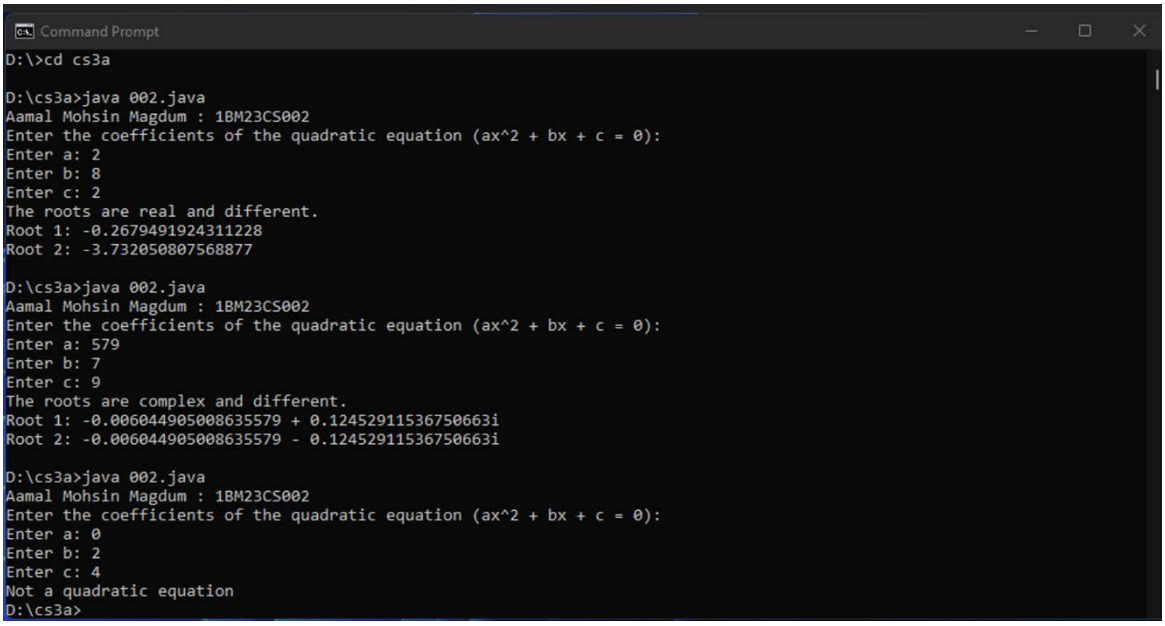
Code:

```
import java.util.Scanner;

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

   Scanner scanner = new Scanner(System.in);

    System.out.println("Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0): ");
```

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

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

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

    if (a==0){
        System.out.print("Not a quadratic equation");
    }
    else{

    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 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 different");
        System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
        System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");
    }
    }
    scanner.close();
}
```

}

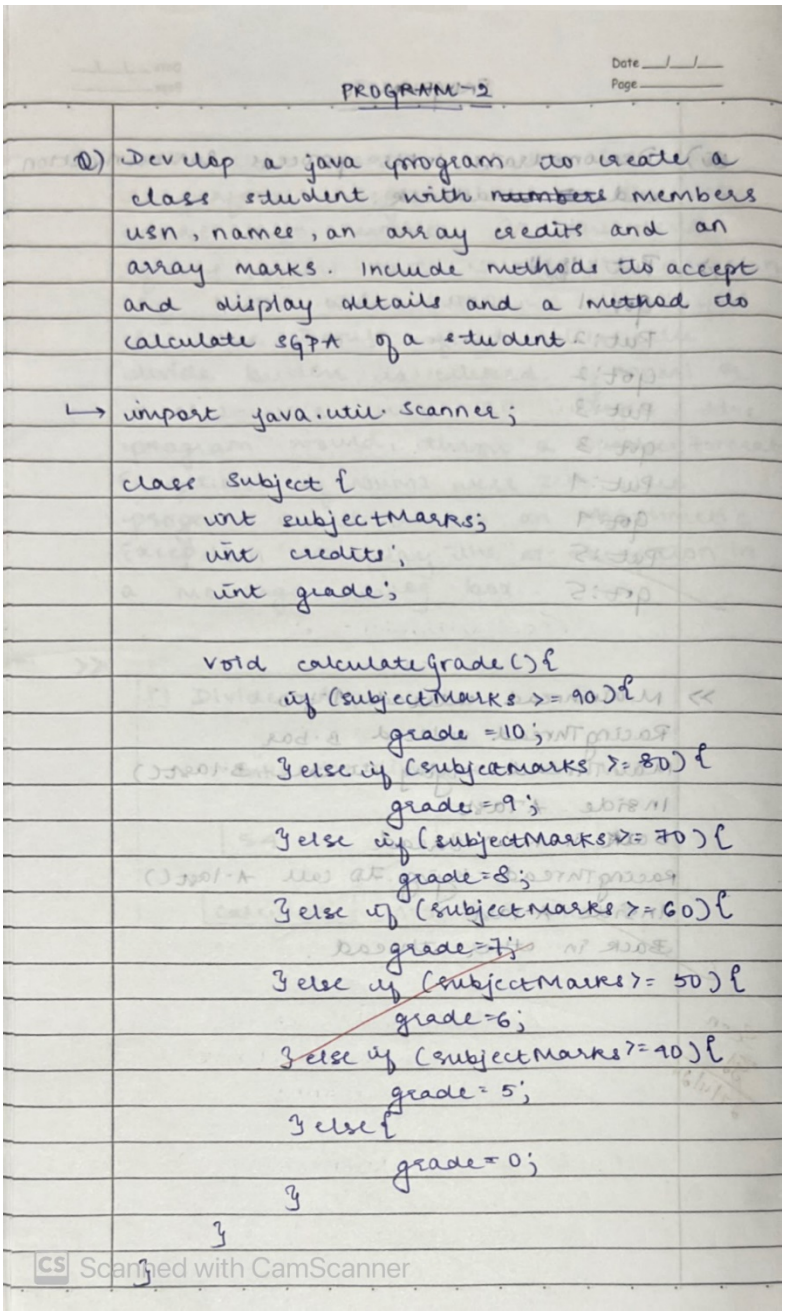
Output:


Command Prompt
D:\>cd cs3a
D:\cs3a>java 002.java
Aamal Mohsin Magdum : 1BM23CS002
Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0):
Enter a: 2
Enter b: 8
Enter c: 2
The roots are real and different.
Root 1: -0.2679491924311228
Root 2: -3.732050807568877
D:\cs3a>java 002.java
Aamal Mohsin Magdum : 1BM23CS002
Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0):
Enter a: 579
Enter b: 7
Enter c: 9
The roots are complex and different.
Root 1: -0.006044905008635579 + 0.12452911536750663i
Root 2: -0.006044905008635579 - 0.12452911536750663i
D:\cs3a>java 002.java
Aamal Mohsin Magdum : 1BM23CS002
Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0):
Enter a: 0
Enter b: 2
Enter c: 4
Not a quadratic equation
D:\cs3a>

Program 2

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

Algorithm:



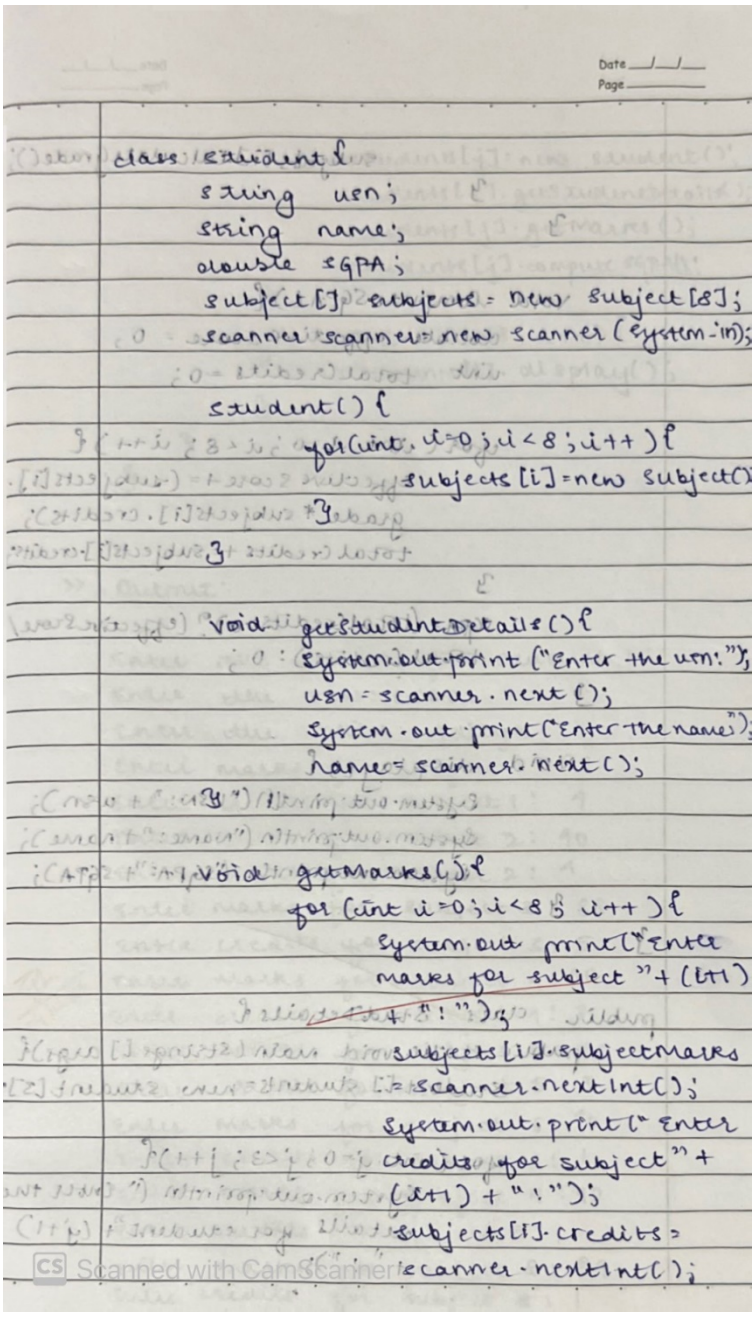
PROGRAM-2

Date __/__/__
Page

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

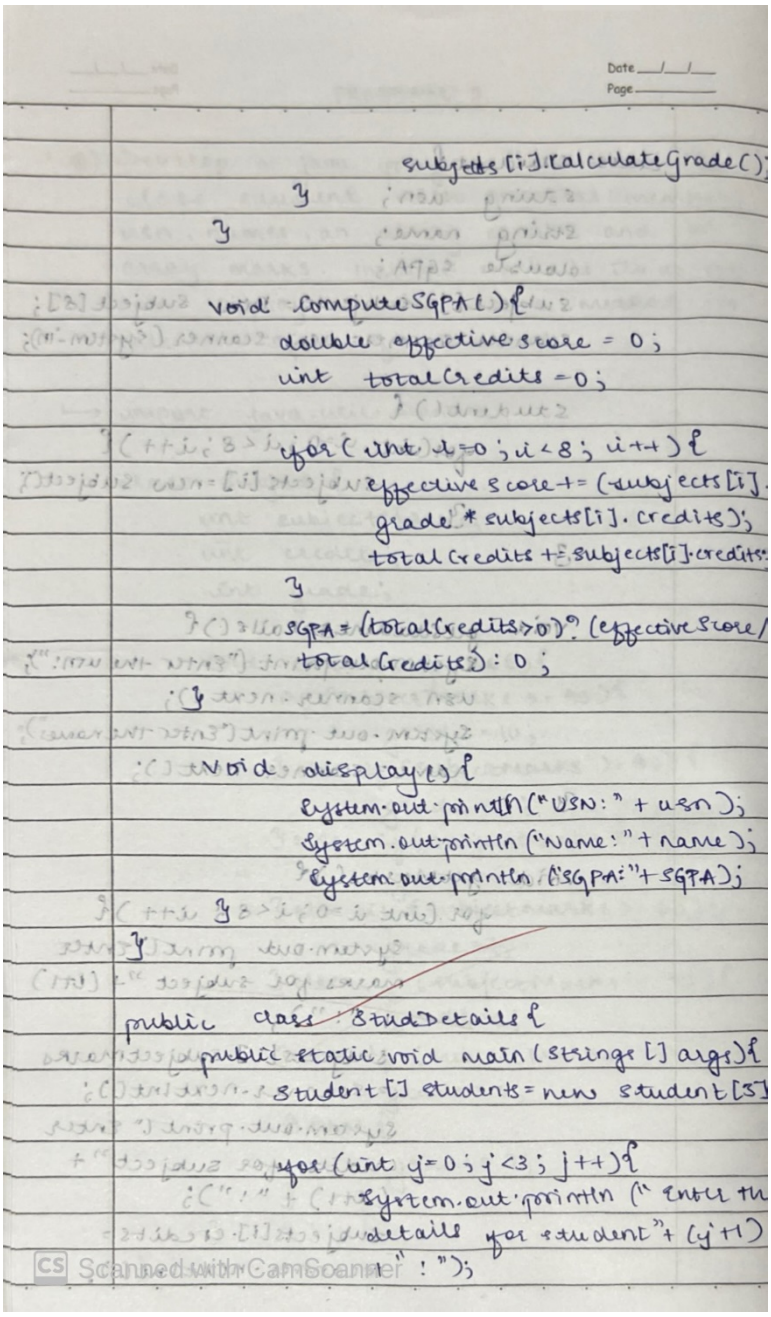
```
import java.util.Scanner;

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

    void calculateGrade() {
        if (subjectMarks >= 90) {
            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;
        }
    }
}
```

CS Scanned with CamScanner



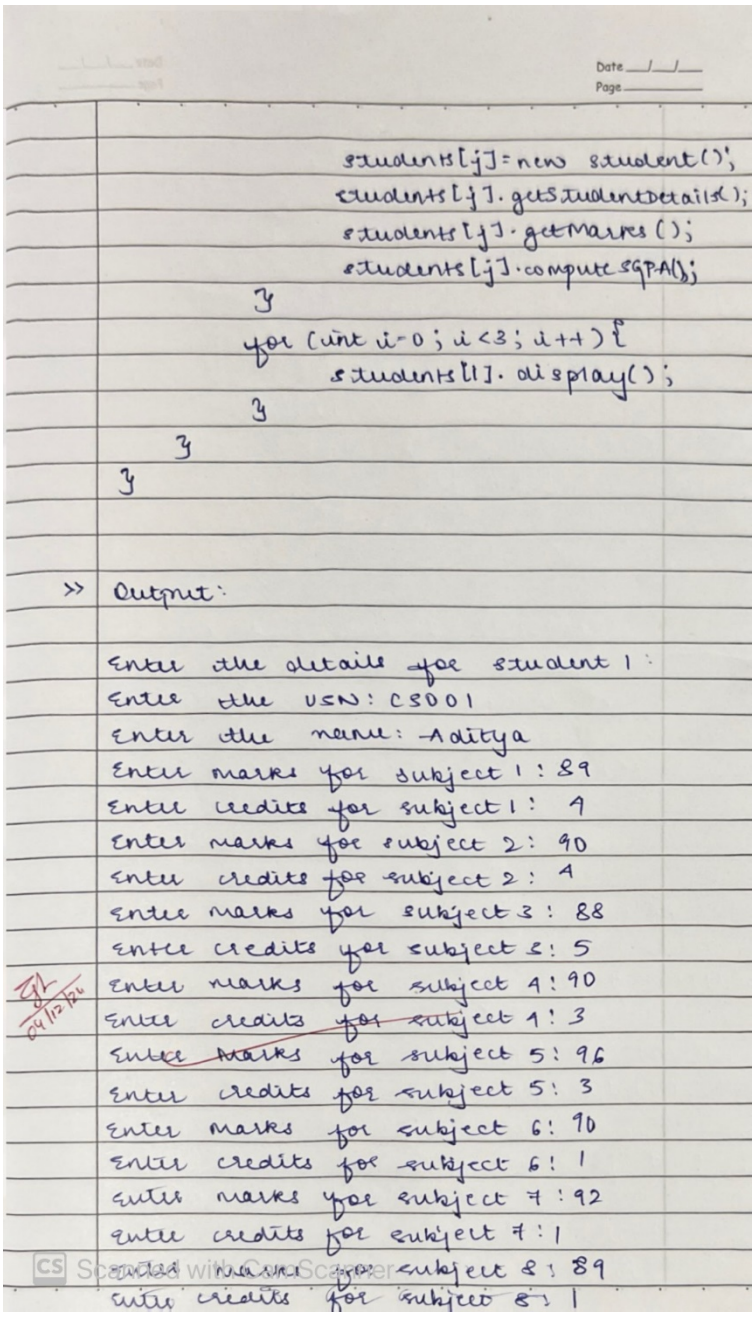
```
Date __/__/__
Page _____

class Student {
    string usn;
    string name;
    double sGPA;
    subject[] subjects = new subject[8];
    scanner scanner= new scanner (System.in);

    student() {
        for(int i=0; i<8; i++) {
            subjects[i]=new subject()
        }
    }

    void getStudentDetails() {
        System.out.print ("Enter the usn:");
        usn = scanner.next();
        System.out.print("Enter the name");
        name= scanner.next();

    void getMarks() {
        for (int i=0; i<8 ; i++) {
            System.out.print("Enter
            marks for subject" + (i+1)
            + ":");
            subjects[i].subjectMarks
            =scanner.nextInt();
            System.out.print(" Enter
            credits for subject" +
            (i+1) + ":");
            subjects[i].credits =
            scanner.nextInt();
```

CS Scanned with CamScanner



```
        subjects[i].calculateGrade();
    }
}
void computeSGPA(){
    double effectiveScore = 0;
    int totalCredits = 0;
    for(int i=0; i<8; i++){
        effectiveScore += (subjects[i].
            grade * subjects[i].credits);
        totalCredits += subjects[i].credits;
    }
    SGPA = (totalCredits>0)? (effectiveScore/
        totalCredits) : 0;
}
void display(){
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("SGPA: "+SGPA);
}
public class StudDetails{
    public static void main(String[] args){
        Student[] students = new Student[5];
        for(int j=0; j<3; j++){
            System.out.println("Enter the
                details for student"+(j+1)
                + " : ");
```

CS Scanned with CamScanner



```
            students[j] = new student();
            students[j].getStudentDetails();
            students[j].getMarks();
            students[j].computeSGPA();
        }
        for (int i=0; i<3; i++) {
            students[i].display();
        }
    }
}
```

Output:

```
Enter the details for student 1:
Enter the USN: CS001
Enter the name: Aditya
Enter marks for subject 1: 89
Enter credits for subject 1: 4
Enter marks for subject 2: 90
Enter credits for subject 2: 4
Enter marks for subject 3: 88
Enter credits for subject 3: 5
Enter marks for subject 4: 90
Enter credits for subject 4: 3
Enter marks for subject 5: 96
Enter credits for subject 5: 3
Enter marks for subject 6: 90
Enter credits for subject 6: 1
Enter marks for subject 7: 92
Enter credits for subject 7: 1
Enter marks for subject 8: 89
Enter credits for subject 8: 1
```

Scanned with CamScanner

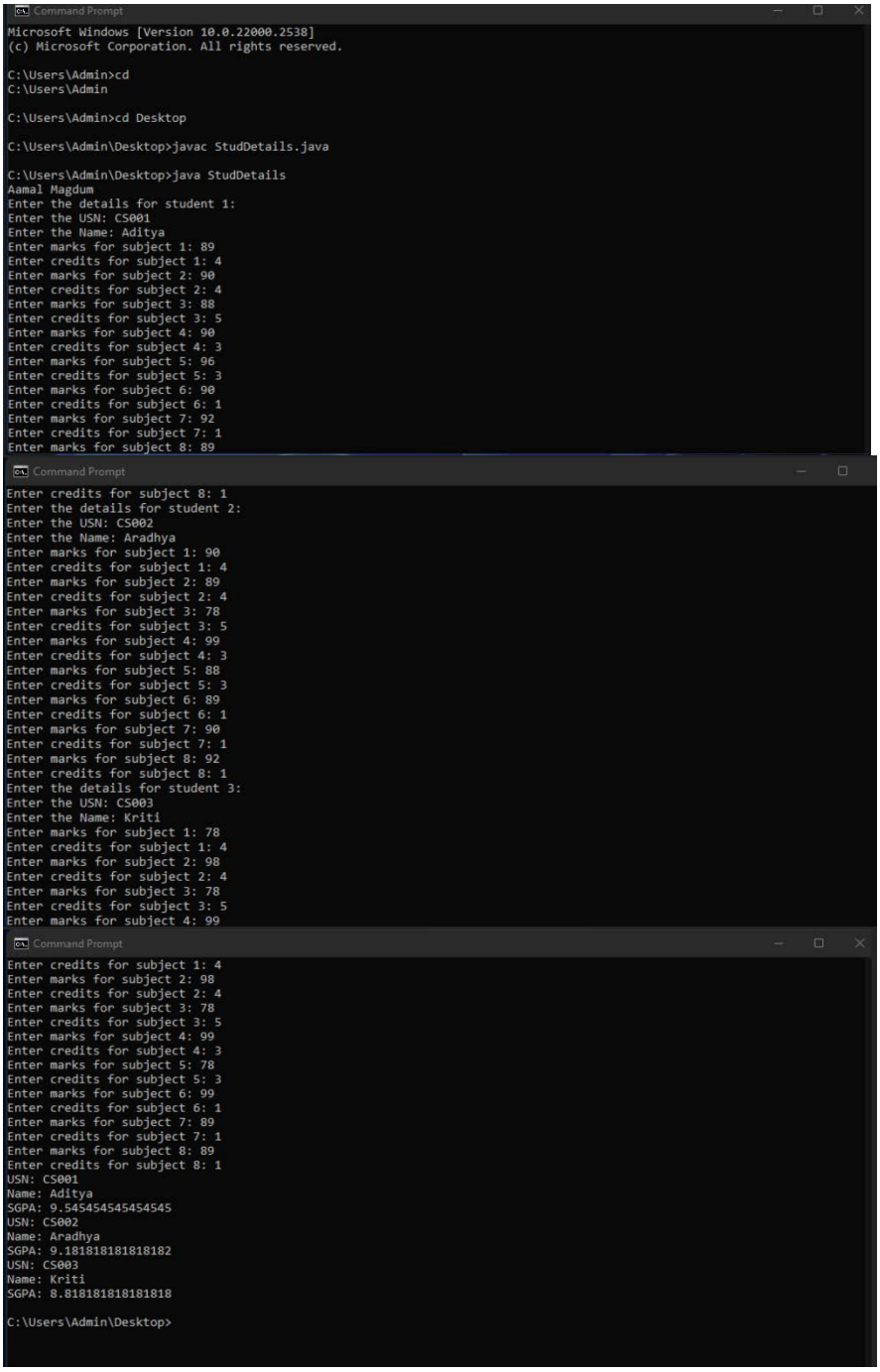
Code:

```
import java.util.Scanner;

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

    void calculateGrade() {
```

```
        if (subjectMarks >= 90) {
            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 usn;
    String name;
    double SGPA;
    Subject[] subjects = new Subject[8];
    Scanner scanner = new Scanner(System.in);

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

    void getStudentDetails() {
        System.out.print("Enter the USN: ");
        usn = scanner.next();
        System.out.print("Enter the Name: ");
        name = scanner.next();
    }

    void getMarks() {
        for (int i = 0; i < 8; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            subjects[i].subjectMarks = scanner.nextInt();
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            subjects[i].credits = scanner.nextInt();
            subjects[i].calculateGrade();
        }
    }
```

```
    void computeSGPA() {
        double effectiveScore = 0;
        int totalCredits = 0;

        for (int i = 0; i < 8; i++) {
            effectiveScore += (subjects[i].grade * subjects[i].credits);
            totalCredits += subjects[i].credits;
        }
        SGPA = (totalCredits > 0) ? (effectiveScore / totalCredits) : 0;
    }

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

public class StudDetails {
    public static void main(String[] args) {
System.out.println("aaryan prakash");
        Student[] students = new Student[3];

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

        for (int i=0;i<3;i++) {
            students[i].display();
        }
    }
}
```

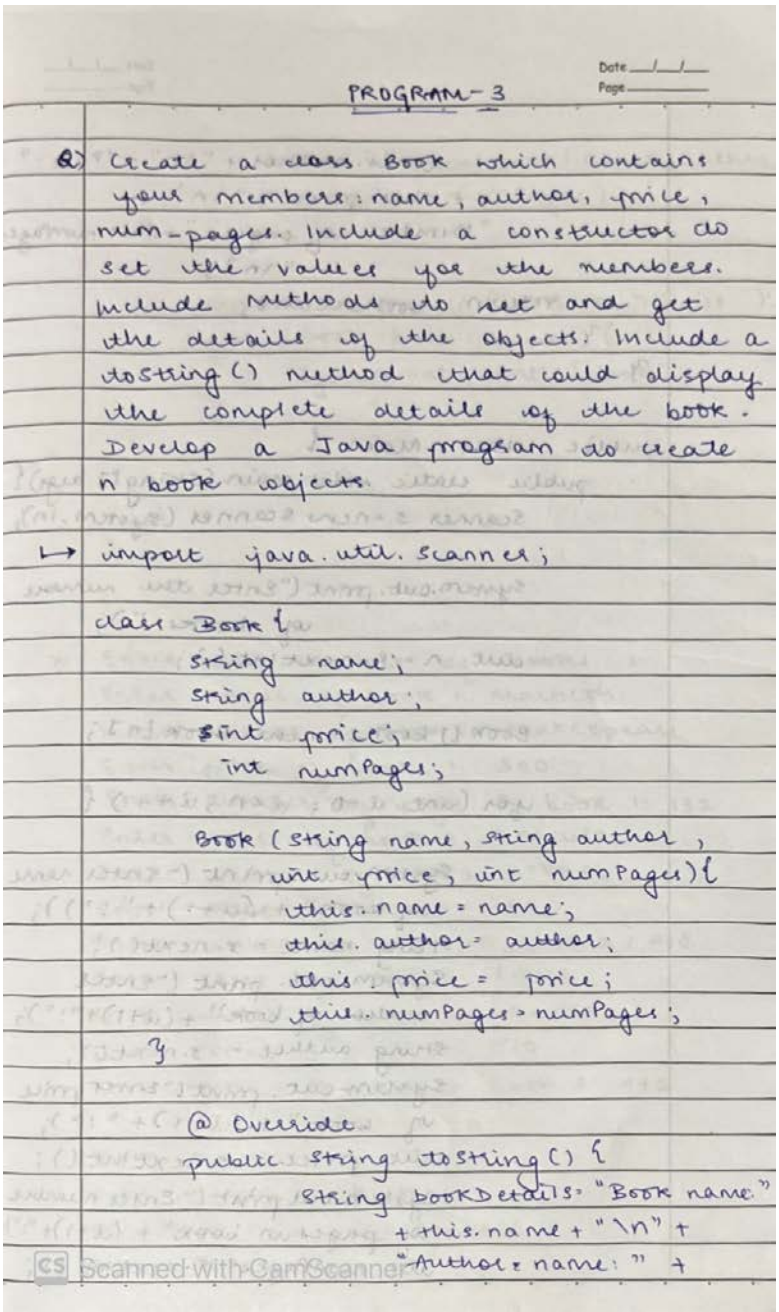
Output:


Command Prompt
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Admin>cd
C:\Users\Admin
C:\Users\Admin>cd Desktop
C:\Users\Admin\Desktop>javac StudDetails.java
C:\Users\Admin\Desktop>java StudDetails
Aamal Magdum
Enter the details for student 1:
Enter the USN: CS001
Enter the Name: Aditya
Enter marks for subject 1: 89
Enter credits for subject 1: 4
Enter marks for subject 2: 90
Enter credits for subject 2: 4
Enter marks for subject 3: 88
Enter credits for subject 3: 5
Enter marks for subject 4: 90
Enter credits for subject 4: 3
Enter marks for subject 5: 96
Enter credits for subject 5: 3
Enter marks for subject 6: 90
Enter credits for subject 6: 1
Enter marks for subject 7: 92
Enter credits for subject 7: 1
Enter marks for subject 8: 89
Command Prompt
Enter credits for subject 8: 1
Enter the details for student 2:
Enter the USN: CS002
Enter the Name: Aradhya
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: 5
Enter marks for subject 4: 99
Enter credits for subject 4: 3
Enter marks for subject 5: 88
Enter credits for subject 5: 3
Enter marks for subject 6: 89
Enter credits for subject 6: 1
Enter marks for subject 7: 90
Enter credits for subject 7: 1
Enter marks for subject 8: 92
Enter credits for subject 8: 1
Enter the details for student 3:
Enter the USN: CS003
Enter the Name: Kriti
Enter marks for subject 1: 78
Enter credits for subject 1: 4
Enter marks for subject 2: 98
Enter credits for subject 2: 4
Enter marks for subject 3: 78
Enter credits for subject 3: 5
Enter marks for subject 4: 99
Command Prompt
Enter credits for subject 1: 4
Enter marks for subject 2: 98
Enter credits for subject 2: 4
Enter marks for subject 3: 78
Enter credits for subject 3: 5
Enter marks for subject 4: 99
Enter credits for subject 4: 3
Enter marks for subject 5: 78
Enter credits for subject 5: 3
Enter marks for subject 6: 99
Enter credits for subject 6: 1
Enter marks for subject 7: 89
Enter credits for subject 7: 1
Enter marks for subject 8: 89
Enter credits for subject 8: 1
USN: CS001
Name: Aditya
SGPA: 9.545454545454545
USN: CS002
Name: Aradhya
SGPA: 9.181818181818182
USN: CS003
Name: Kriti
SGPA: 8.818181818181818
C:\Users\Admin\Desktop>

Program 3

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

Algorithm:



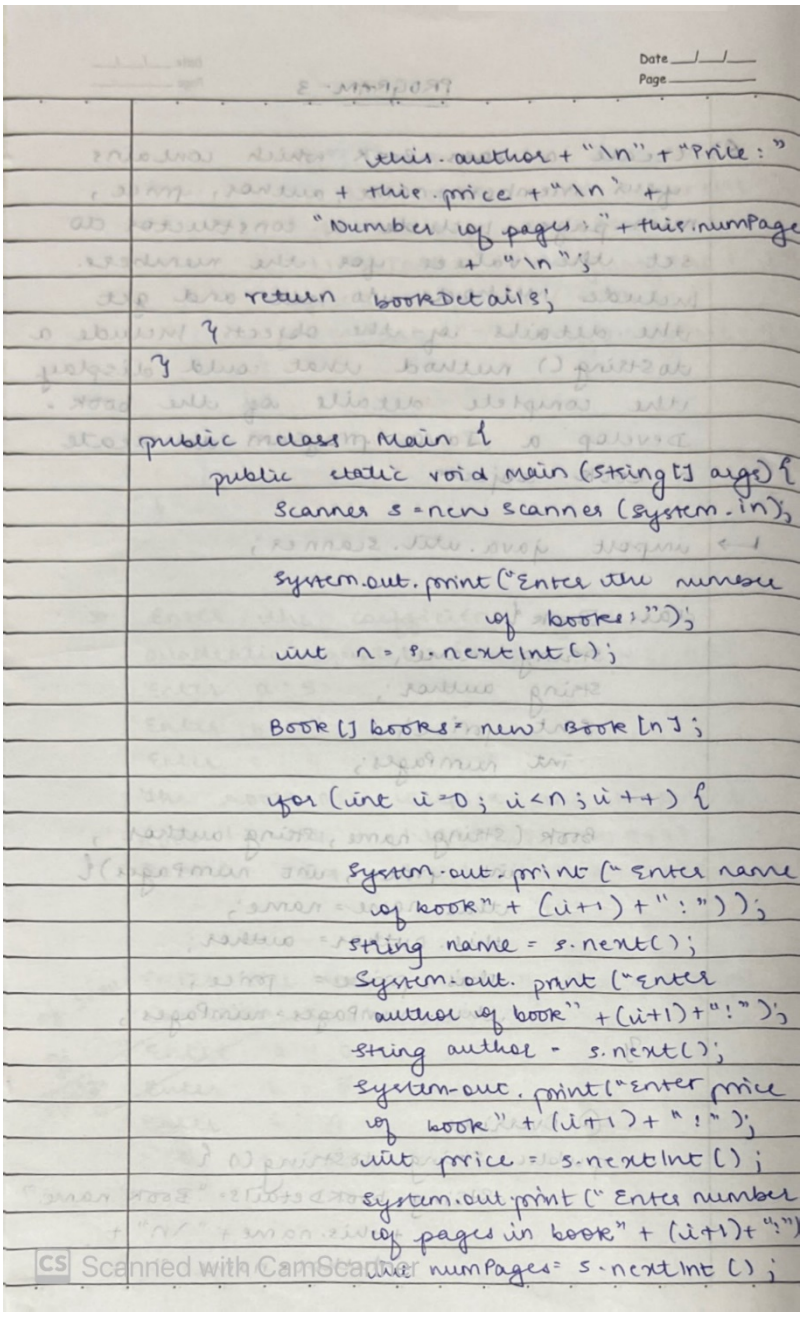
PROGRAM-3

Date __/__/__
Page _____

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

```
import java.util.Scanner;

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

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

    @Override
    public String toString() {
        String bookDetails = "Book name:"
            + this.name + "\n" +
            "Author name: " +
```

Scanned with CamScanner



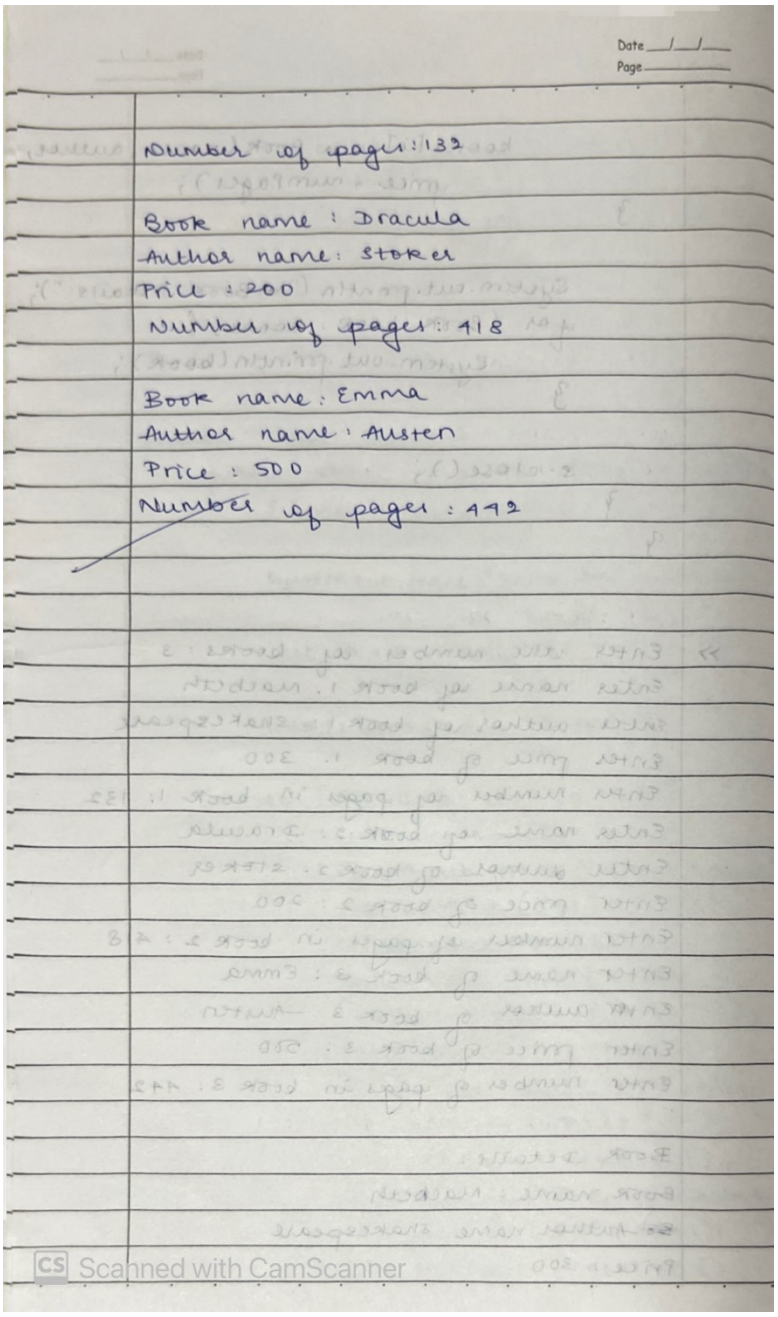
```
Date __/__/____
Page _____

        this.author + "\n" + "Price: "
        + this.price + "\n" +
        "Number of pages: " + this.numPage
        + "\n";
        return bookDetails;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the number
                of books:");
        int n = s.nextInt();

        Book[] books = new Book[n];

        for (int i=0; i<n; i++) {
            System.out.print("Enter name
                of book" + (i+1) + ":");
            String name = s.next();
            System.out.print("Enter
                author of book" + (i+1) + ":");
            String author = s.next();
            System.out.print("Enter price
                of book" + (i+1) + ":");
            int price = s.nextInt();
            System.out.print("Enter number
                of pages in book" + (i+1) + ":")
            int numPages = s.nextInt();
```

CS Scanned with CamScanner

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

        System.out.println("\n Book Details:");
        for (Book book : books) {
            System.out.println(book);
        }

        s.close();
    }
}
```

```
» Enter the number of books : 3
Enter name of book 1: Macbeth
Enter author of book 1: shakespeare
Enter price of book 1: 300
Enter number of pages in book 1: 132
Enter name of book 2: Dracula
Enter author of book 2: stoker
Enter price of book 2: 200
Enter number of pages in book 2: 418
Enter name of book 3: Emma
Enter author of book 3: Austen
Enter price of book 3: 500
Enter number of pages in book 3: 412

Book Details:
Book name: Macbeth
Author name: shakespeare
Price: 300
```

CS Scanned with CamScanner

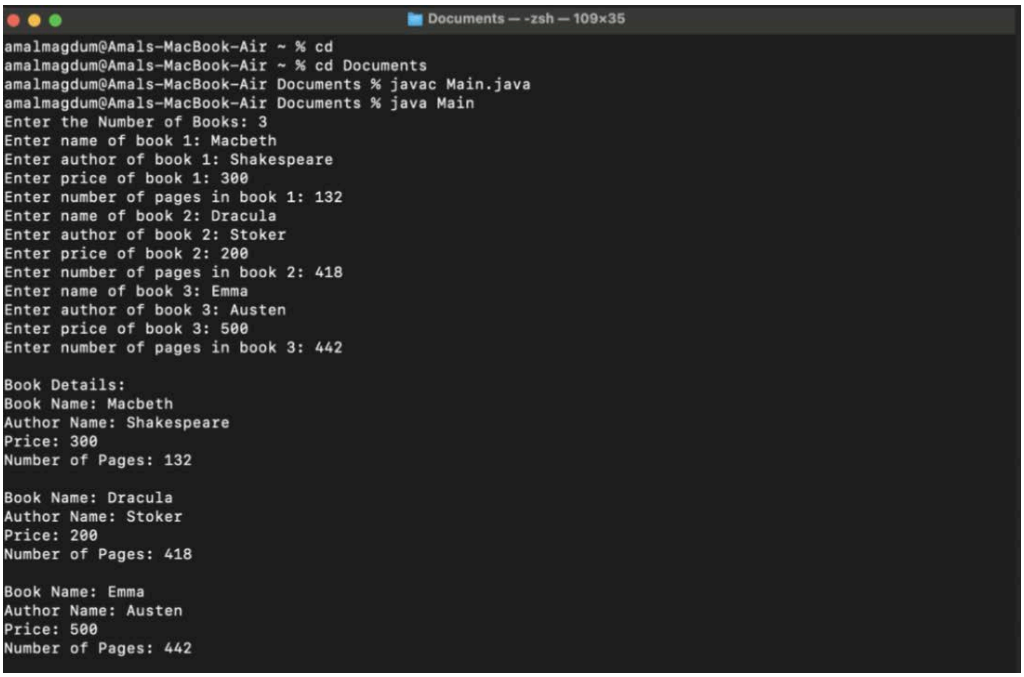


Code:

```
import java.util.Scanner;

class Book {
    String name;
    String author;
    int price;
    int numPages;
```

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

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

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

        System.out.print("Enter the Number of Books: ");
        int n = s.nextInt();

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter name of book " + (i + 1) + ": ");
            String name = s.next();
            System.out.print("Enter author of book " + (i + 1) + ": ");
            String author = s.next();
            System.out.print("Enter price of book " + (i + 1) + ": ");
            int price = s.nextInt();
            System.out.print("Enter number of pages in book " + (i + 1) + ": ");
            int numPages = s.nextInt();

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

        System.out.println("\nBook Details:");
        for (Book book : books) {
            System.out.println(book);
        }

        s.close();
    }
}
```

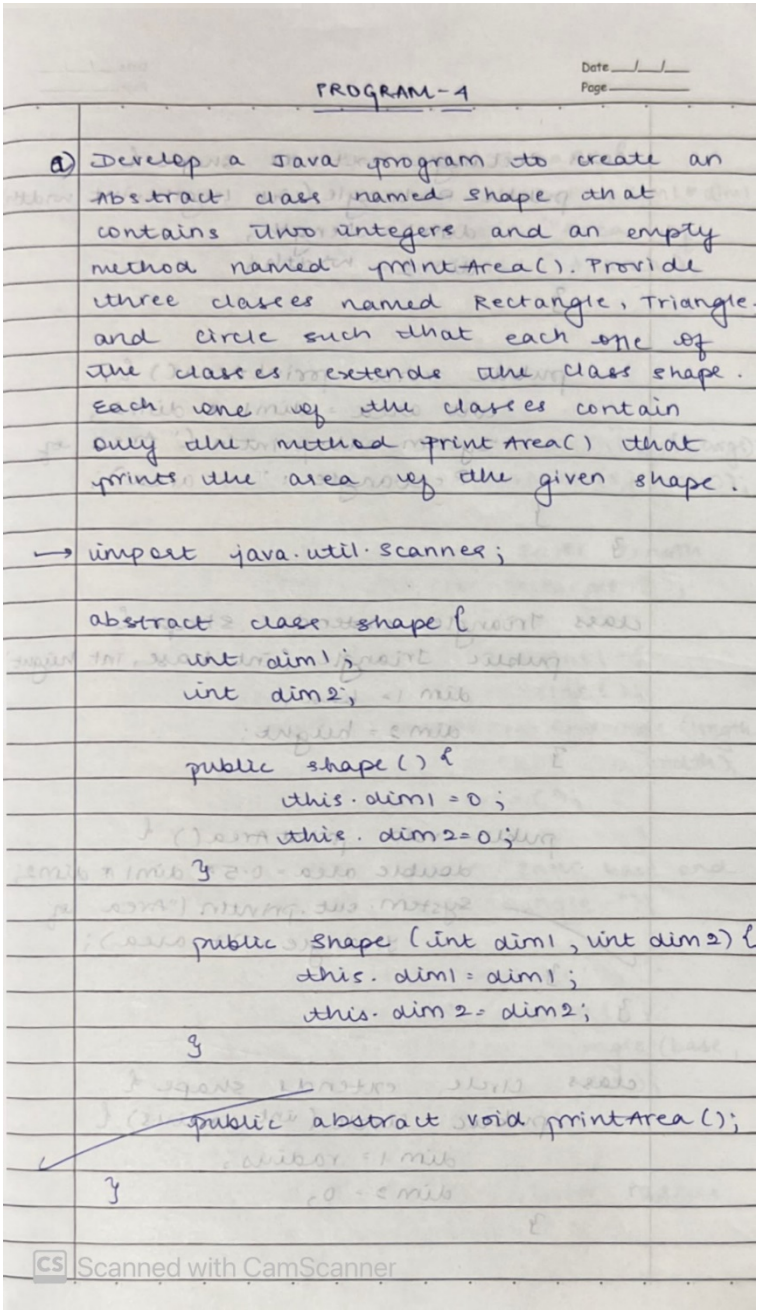
Output:


Documents — -zsh — 109×35
amalmagdum@Amals-MacBook-Air ~ % cd
amalmagdum@Amals-MacBook-Air ~ % cd Documents
amalmagdum@Amals-MacBook-Air Documents % javac Main.java
amalmagdum@Amals-MacBook-Air Documents % java Main
Enter the Number of Books: 3
Enter name of book 1: Macbeth
Enter author of book 1: Shakespeare
Enter price of book 1: 300
Enter number of pages in book 1: 132
Enter name of book 2: Dracula
Enter author of book 2: Stoker
Enter price of book 2: 200
Enter number of pages in book 2: 418
Enter name of book 3: Emma
Enter author of book 3: Austen
Enter price of book 3: 500
Enter number of pages in book 3: 442
Book Details:
Book Name: Macbeth
Author Name: Shakespeare
Price: 300
Number of Pages: 132
Book Name: Dracula
Author Name: Stoker
Price: 200
Number of Pages: 418
Book Name: Emma
Author Name: Austen
Price: 500
Number of Pages: 442

Program 4

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

Algorithm:



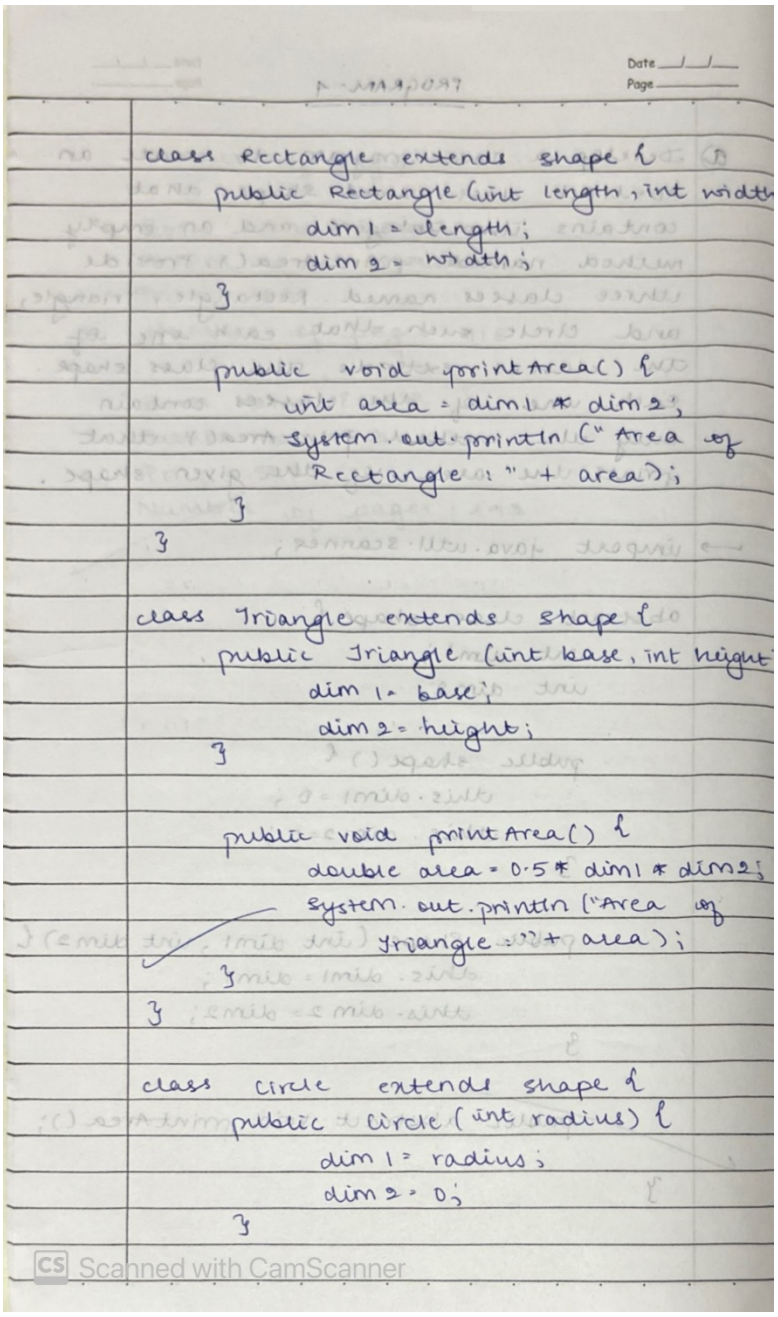
PROGRAM-4

Date __/__/__
Page _____

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

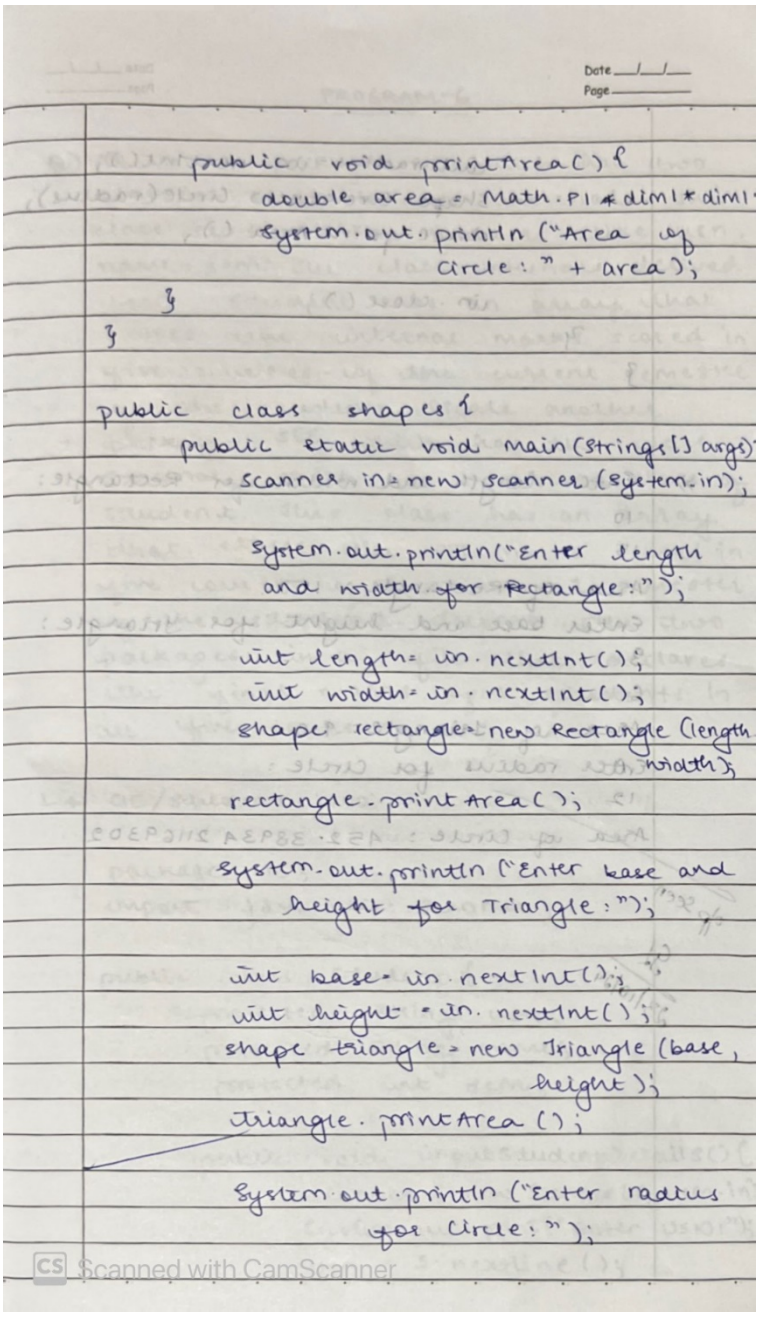
```
import java.util.scanner;

abstract class shape {
    int dim1;
    int dim2;

    public shape() {
        this.dim1 = 0;
        this.dim2 = 0;
    }

    public shape (int dim1, int dim2) {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }

    public abstract void printArea();

}
```

Scanned with CamScanner



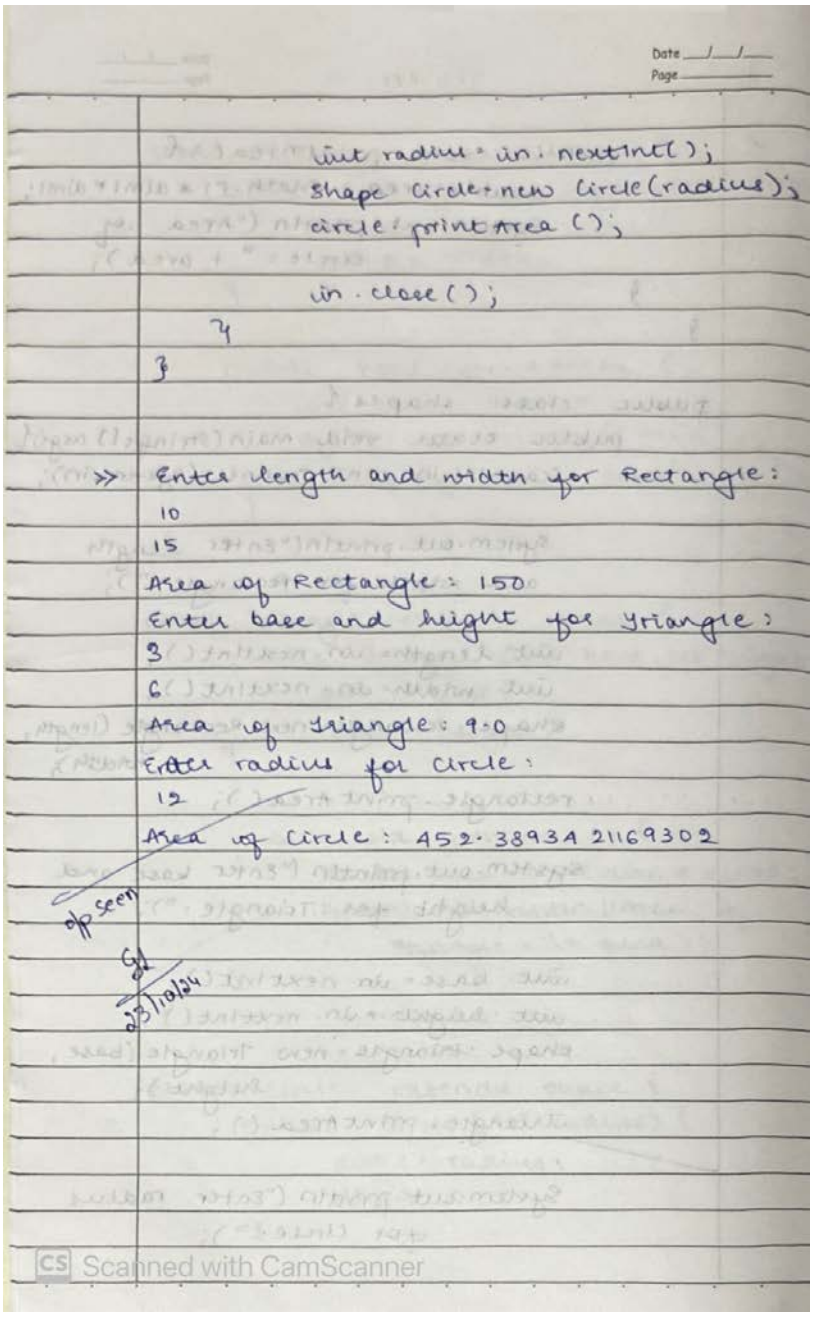
```
class Rectangle extends shape {
    public Rectangle (int length, int width
        dim1 = length;
        dim2 = width;
    }
    public void printArea() {
        int area = dim1 * dim2;
        System.out.println("Area of
            Rectangle : " + area);
    }
}

class Triangle extends shape {
    public Triangle (int base, int height
        dim1 = base;
        dim2 = height;
    }
    public void printArea() {
        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of
            Triangle :" + area);
    }
}

class Circle extends shape {
    public Circle (int radius) {
        dim1 = radius;
        dim2 = 0;
    }
```

CS Scanned with CamScanner



```
    public void printArea() {
        double area = Math.PI * dim1 * dim1;
        System.out.println("Area of
                circle: " + area);
    }
}

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

        System.out.println("Enter length
            and width for Rectangle:");
        int length = in.nextInt();
        int width = in.nextInt();
        Shape rectangle = new Rectangle(length,
                                        width);
        rectangle.printArea();
        System.out.println("Enter base and
            height for Triangle:");

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

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

CS Scanned with CamScanner



Code:

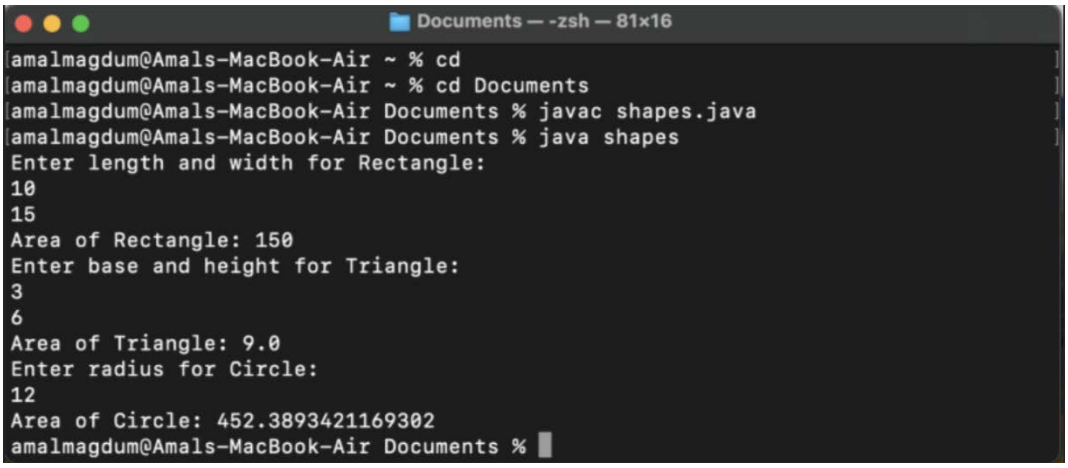
```
import java.util.Scanner;

abstract class Shape {

int dim1;

int dim2;
```

```
public Shape() {

   this.dim1 = 0;

   this.dim2 = 0;

}

public Shape(int dim1, int dim2) {

   this.dim1 = dim1;

   this.dim2 = dim2;

}

public abstract void printArea();


}

class Rectangle extends Shape {

public Rectangle(int length, int width) {
   dim1 = length;
   dim2 = width;
}

public void printArea() {

   int area = dim1 * dim2;
   System.out.println("Area of Rectangle: " + area);

}

}

class Triangle extends Shape {

public Triangle(int base, int height) {

   dim1 = base;
   dim2 = height;

}

public void printArea() {
```

```
    double area = 0.5 * dim1 * dim2;
    System.out.println("Area of Triangle: " + area);

}

}

class Circle extends Shape {

public Circle(int radius) {
    dim1 = radius;
    dim2 = 0;
}

public void printArea() {
    double area = Math.PI * dim1 * dim1;
    System.out.println("Area of Circle: " + area);

}


}

public class shapes {

public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    System.out.println("Enter length and width for Rectangle:");

    int length = in.nextInt();
    int width = in.nextInt();

    Shape rectangle = new Rectangle(length, width);

    rectangle.printArea();

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

    int base = in.nextInt();
    int height = in.nextInt();

    Shape triangle = new Triangle(base, height);

    triangle.printArea();

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

    int radius = in.nextInt();
    Shape circle = new Circle(radius);
```

```
    circle.printArea();

    in.close();

}
}
```

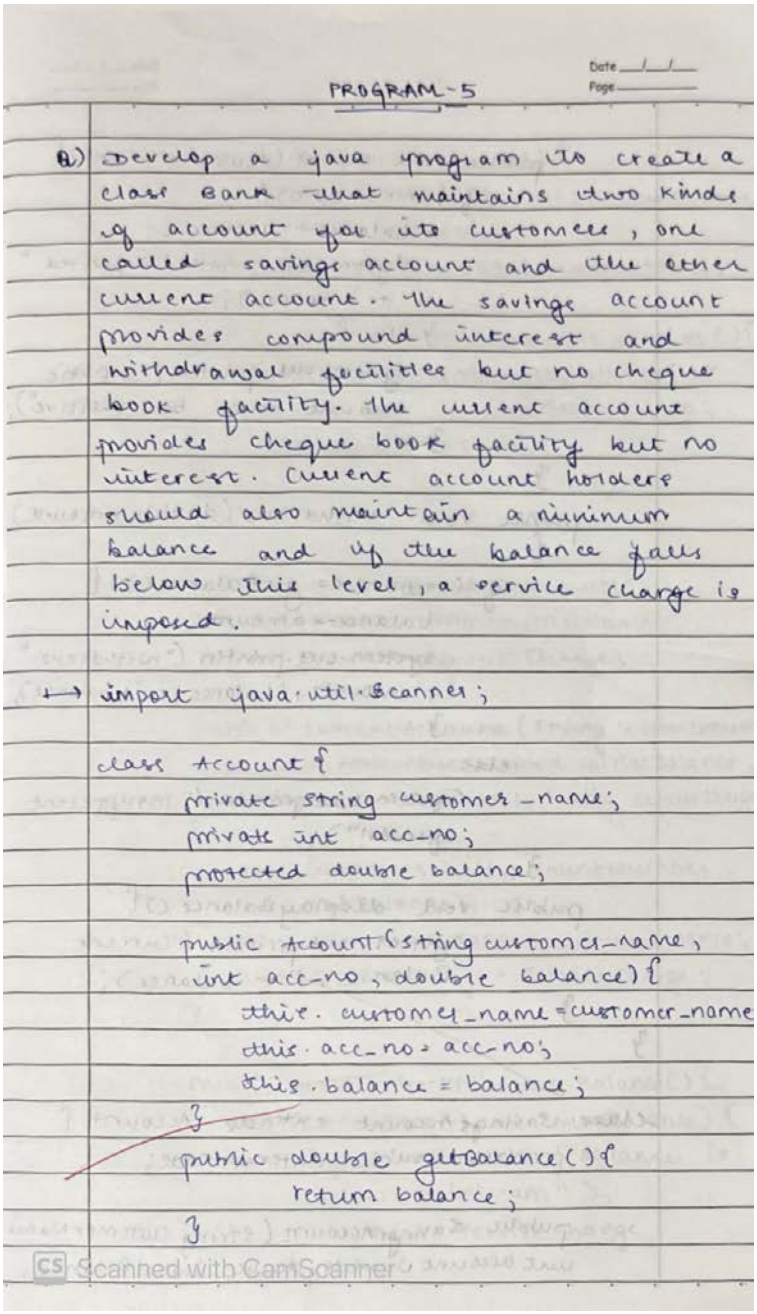
Output:



Program 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Algorithm:

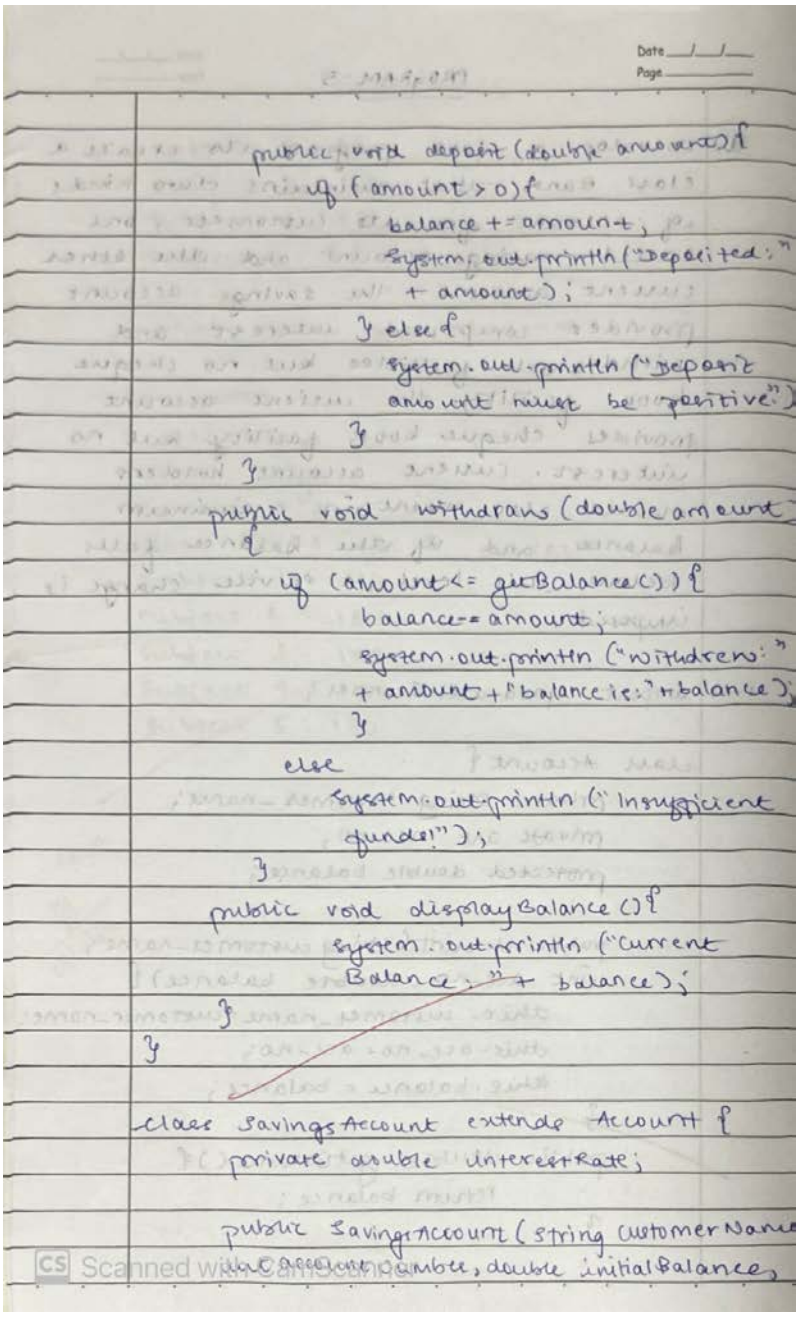


PROGRAM-5

a) Develop a java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

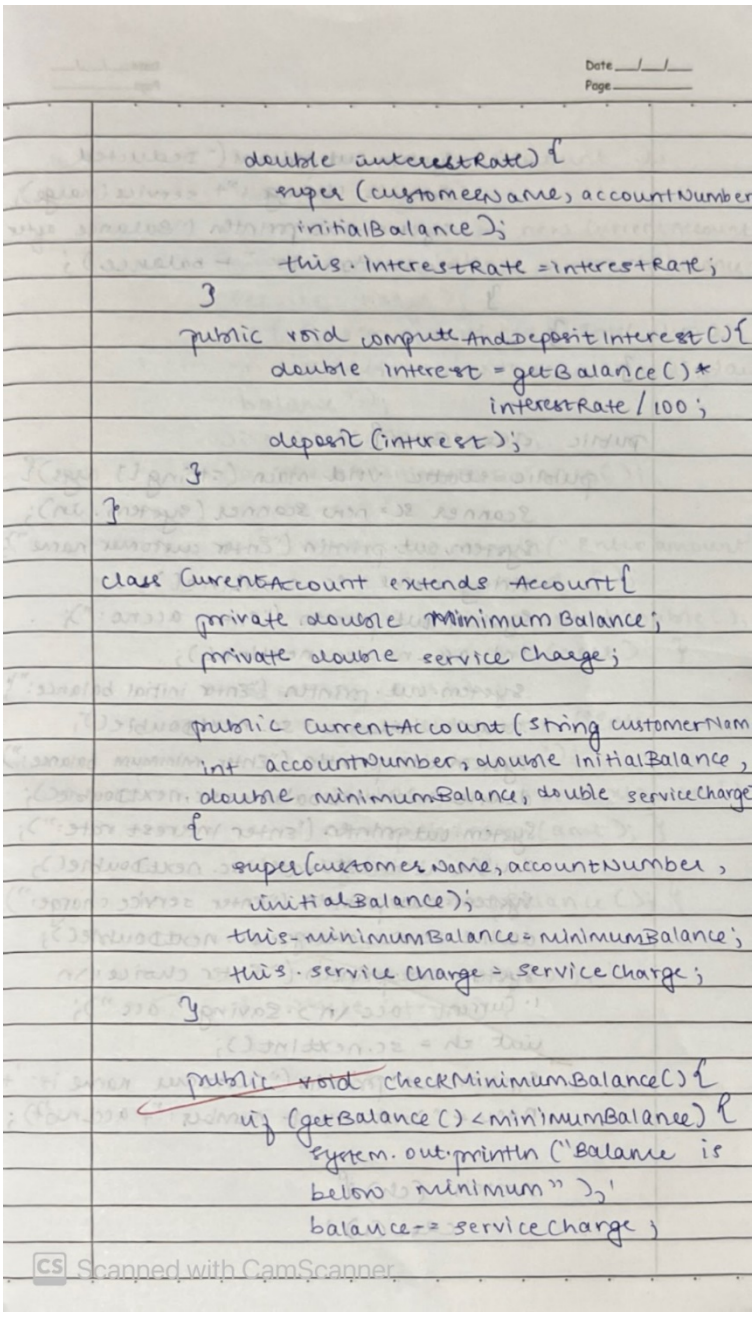
```
import java.util.Scanner;

class Account {
    private String customer_name;
    private int acc_no;
    protected 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;
    }
```

Scanned with CamScanner



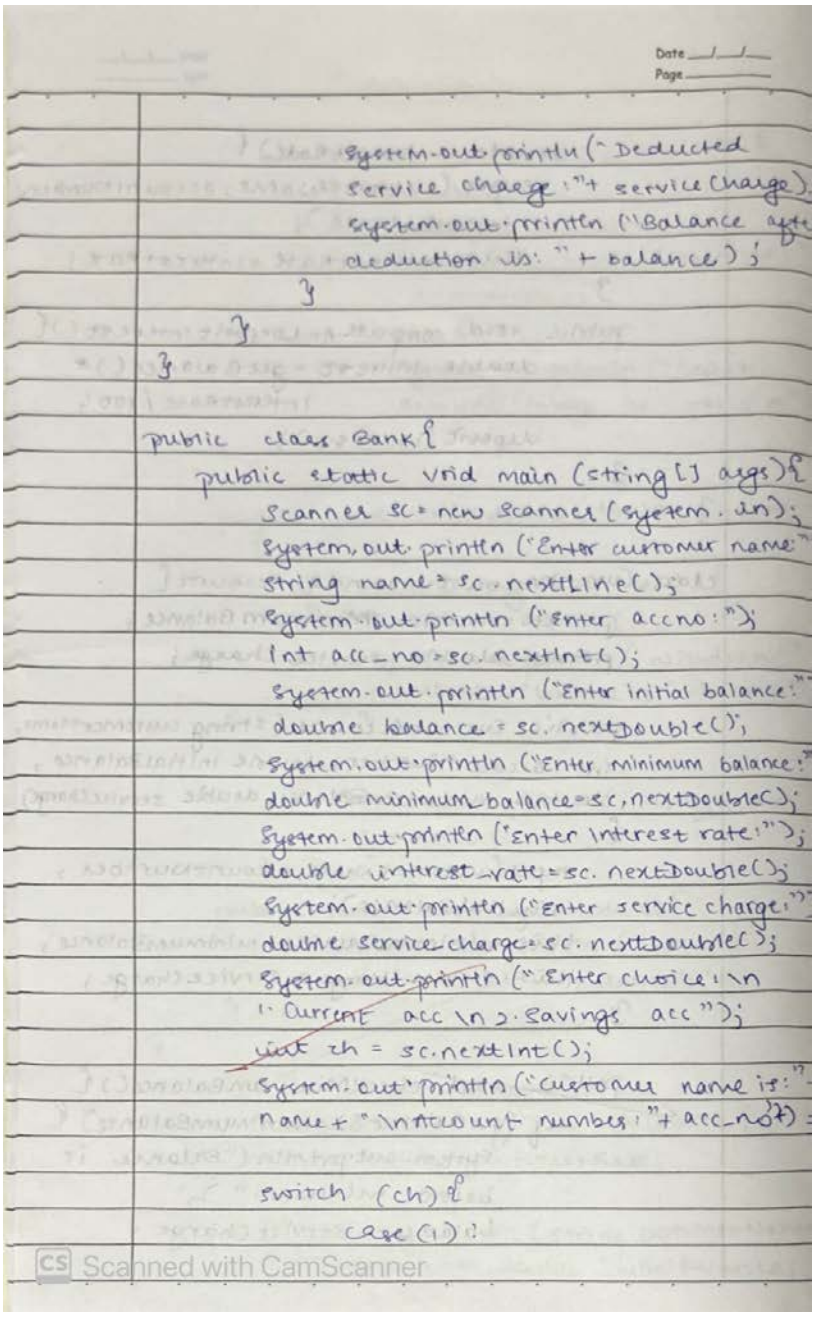
```
public void deposit (double amount){
    if (amount > 0){
        balance += amount;
        System.out.println("Deposited: "
        + amount);
    } else{
        System.out.println("Deposit
        amount must be positive");
    }
}
public void withdraw (double amount)
{
    if (amount <= getBalance()){
        balance -= amount;
        System.out.println("withdrew: "
        + amount + "balance is: " + balance);
    }
    else
        System.out.println("Insufficient
        funds!");
}
public void displayBalance(){
    System.out.println("Current
    Balance: " + balance);
}
}

class SavingsAccount extends Account {
    private double interestRate;

    public SavingsAccount(String customerName
    , String accountNumber, double initialBalance,
```

CS Scanned with CamScanner



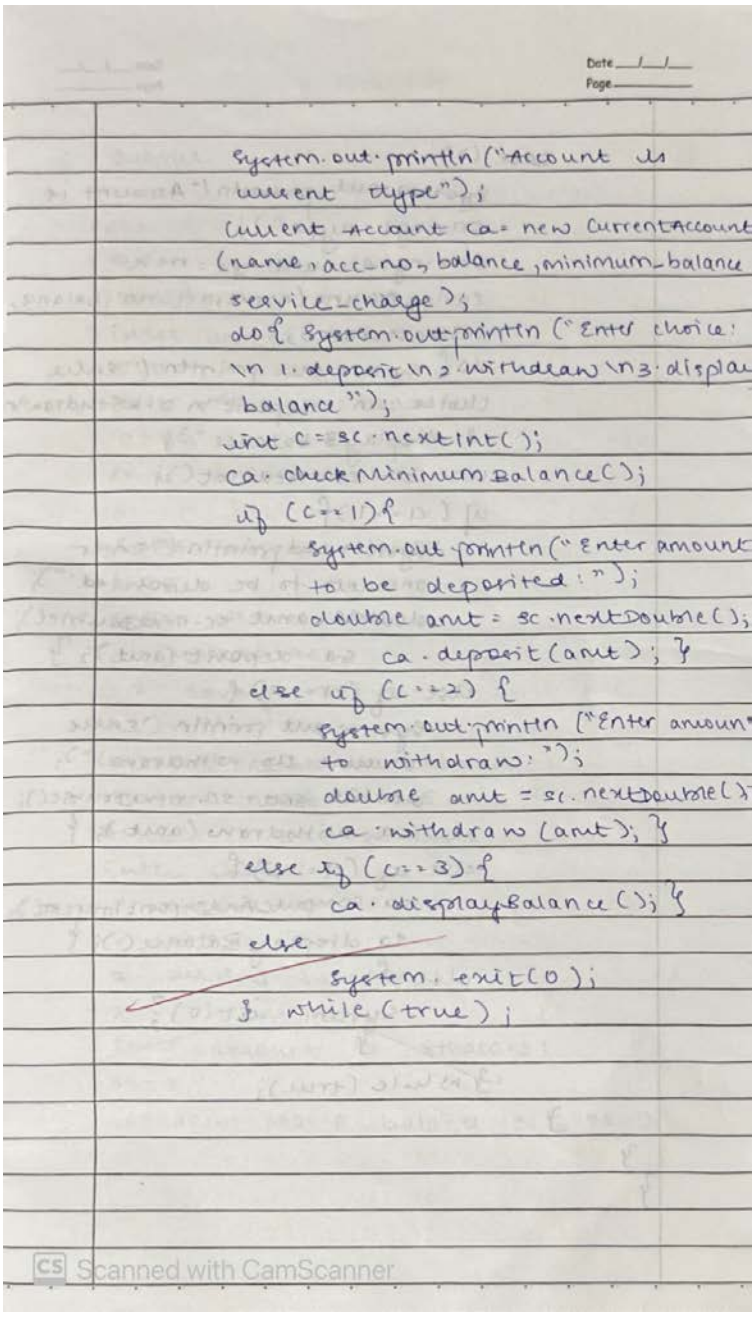
```
        double interestRate) {
            super (customerName, accountNumber
                initialBalance);
            this.interestRate = interestRate;
        }
        public void computeAndDepositInterest(){
            double interest = getBalance() *
                                interestRate / 100;
            deposit (interest);
        }
}
class CurrentAccount extends Account{
    private double minimumBalance;
    private double serviceCharge;

    public CurrentAccount (String customerNam
    int accountNumber, double initialBalance,
    double minimumBalance, double serviceCharge
    {
        super(customerName, accountNumber,
        initialBalance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    public void checkMinimumBalance(){
        if (getBalance() < minimumBalance) {
            System.out.println("Balance is
            below minimum");
            balance -= serviceCharge;
```

CS Scanned with CamScanner



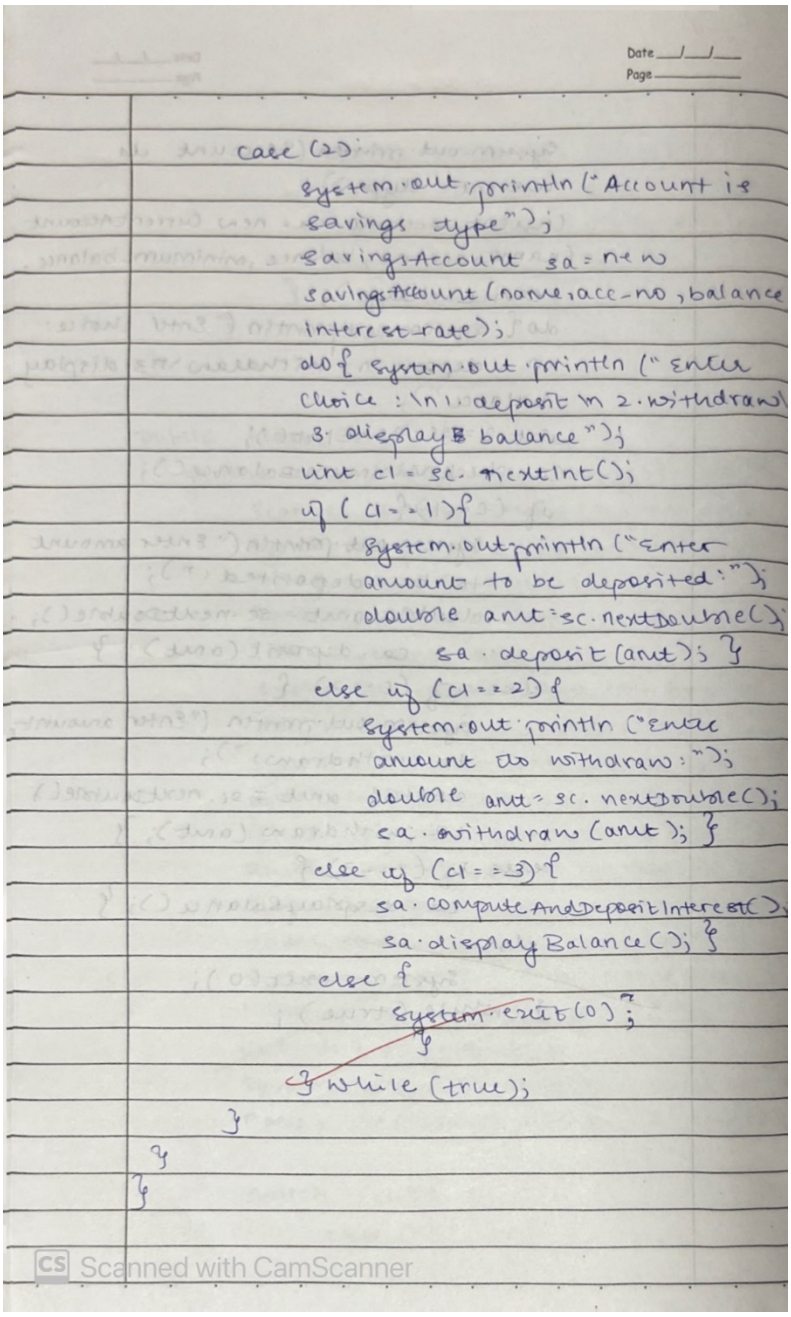
```
            System.out.println("Deducted
            service charge:"+ service charge);
            System.out.println("Balance aft
            deduction is: " + balance);
        }
    }
}

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 service charge:"
        double service_charge=sc.nextDouble();
        System.out.println("Enter choice:\n
        1.Current acc\n2.Savings acc");
        int ch = sc.nextInt();
        System.out.println("Customer name is:"
        name+"\nAccount number:"+acc_no);

        switch (ch){
            case(1):
```

CS Scanned with CamScanner



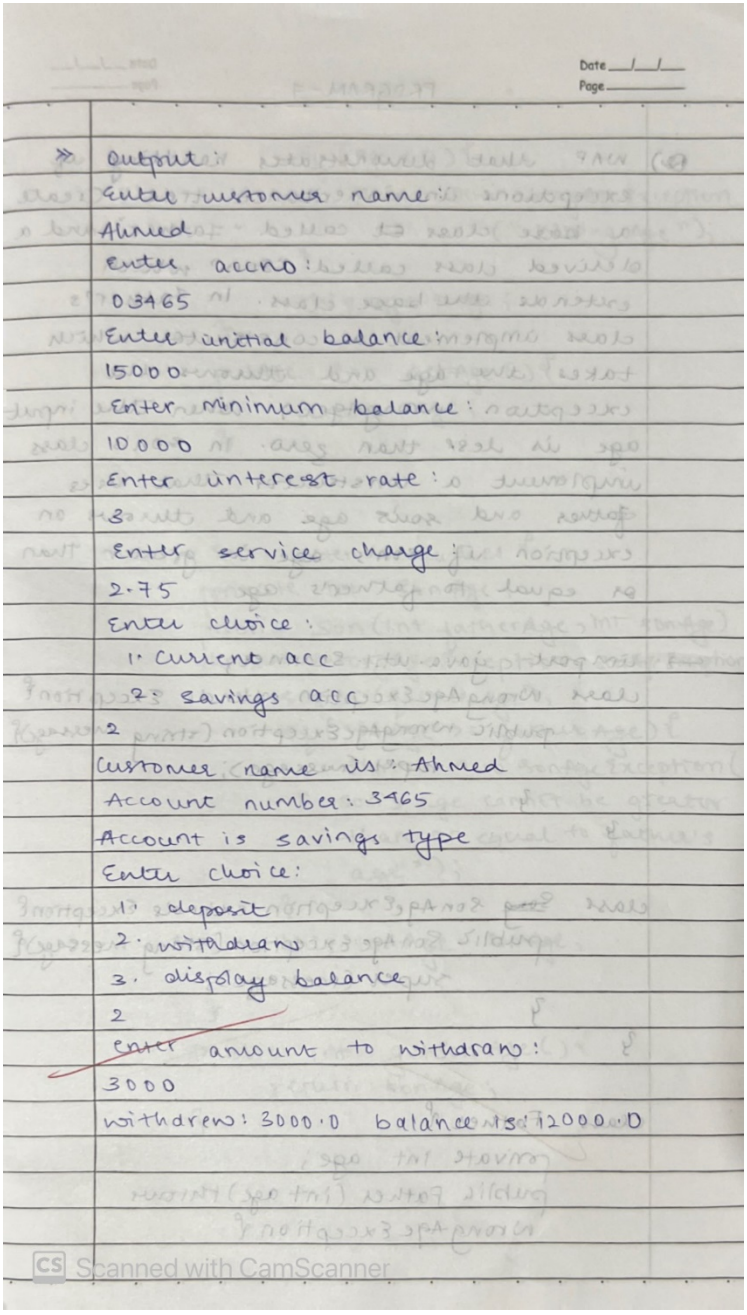
```
System.out.println("Account is
current type");
Current Account ca= new CurrentAccount
(name, acc_no, balance, minimum_balance
service_charge);
do{ System.out.println("Enter choice:
\n 1. deposit \n 2. withdraw \n 3. displa
balance");
int c= sc.nextInt();
ca.checkMinimumBalance();
if (c==1){
    System.out.println("Enter amount
    to be deposited:");
    double amt = sc.nextDouble();
        ca.deposit(amt); }
else if (c==2) {
    System.out.println("Enter amoun
    to withdraw:");
    double amt = sc.nextDouble();
    ca.withdraw(amt); }
else if (c==3) {
    ca.displayBalance(); }
else
    System.exit(0);
} while (true);
```

Scanned with CamScanner



```
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 : \n1. deposit \n 2. withdraw
        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.computeAndDepositInterest();
            sa.displayBalance(); }
        else {
            System.exit(0);
        } while (true);
    }
}
}
```

Scanned with CamScanner



Code:

```
import java.util.Scanner;

class Account { private String customer_name; private int acc_no; protected 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);
   } else {
      System.out.println("Deposit amount must be positive.");
   }
}


public void withdraw(double amount) { if(amount<=getBalance()){ balance-=amount;
System.out.println("withdrew:"+amount + " balance is:"+ balance); } else
System.out.println("Insufficient funds!!"); } 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);
}


} 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 checkMinimumBalance() {
   if (getBalance() < minimumBalance) {
      System.out.println("Balance is below minimum");
      balance-=serviceCharge;
```

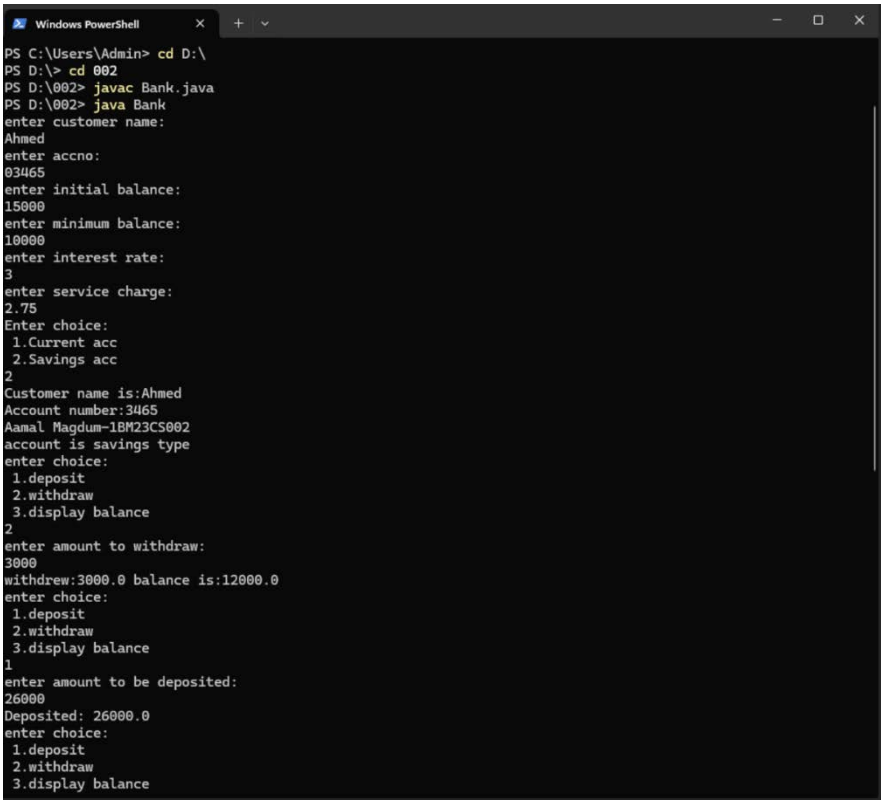
```
        System.out.println("Deducted service charge:" +serviceCharge);
        System.out.println("Balance after deduction is:"+balance);
    }
}
```

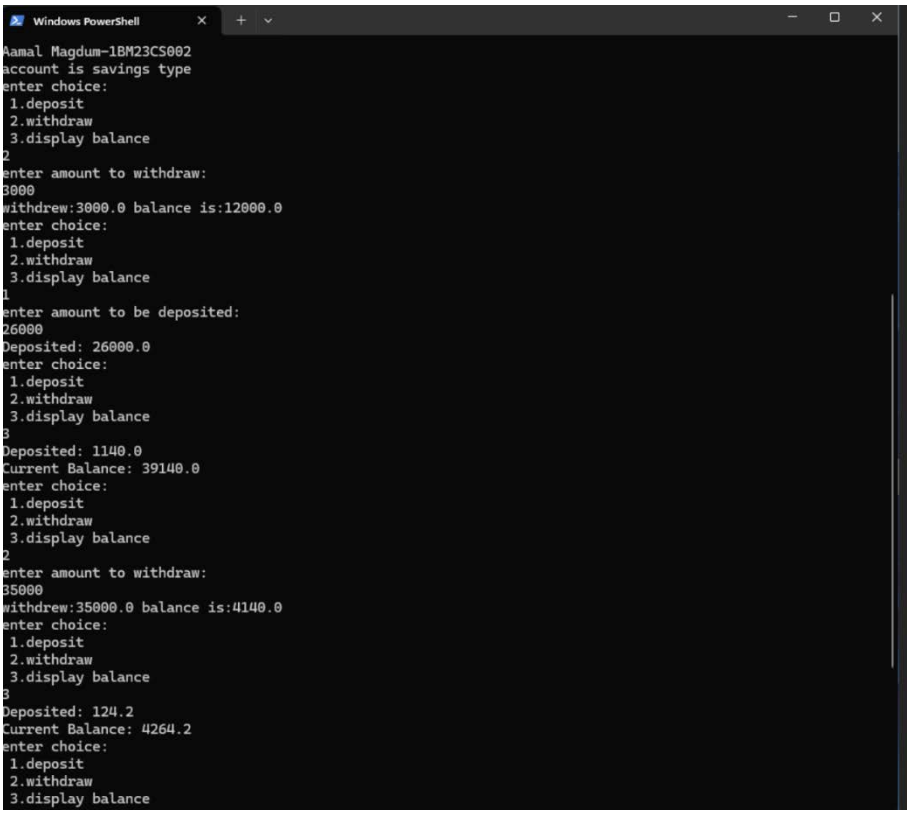
```
} public class Bank { public static void main(String[] args) { Scanner sc = new Scanner(System.in);
System.out.println("enter customer name:"); String name=sc.nextLine(); System.out.println("enter
accno:"); int acc_no=sc.nextInt();

 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 service charge:"); double service_charge=sc.nextDouble();
System.out.println("Enter choice:\n 1.Current acc\n 2.Savings acc"); int ch=sc.nextInt();
System.out.println("Customer name is:"+ name+"\nAccount number:"+acc_no); switch(ch){ case(1):
System.out.println("account is current type"); CurrentAccount ca = new
CurrentAccount(name,acc_no,balance,minimum_balance,service_charge); do{
System.out.println("enter choice:\n 1.deposit\n 2.withdraw\n 3.display balance"); int c=sc.nextInt();
ca.checkMinimumBalance(); if(c==1){ System.out.println("enter amount to be deposited:"); double
amt=sc.nextDouble(); ca.deposit(amt);} else if(c==2){ System.out.println("enter amount to
withdraw:"); double amt=sc.nextDouble(); ca.withdraw(amt);} else if(c==3){ ca.displayBalance();}
else System.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.computeAndDepositInterest();
           sa.displayBalance();}
        else{
          System.exit(0);
             }
        }while(true);
}
```

```
} }
```

Output:

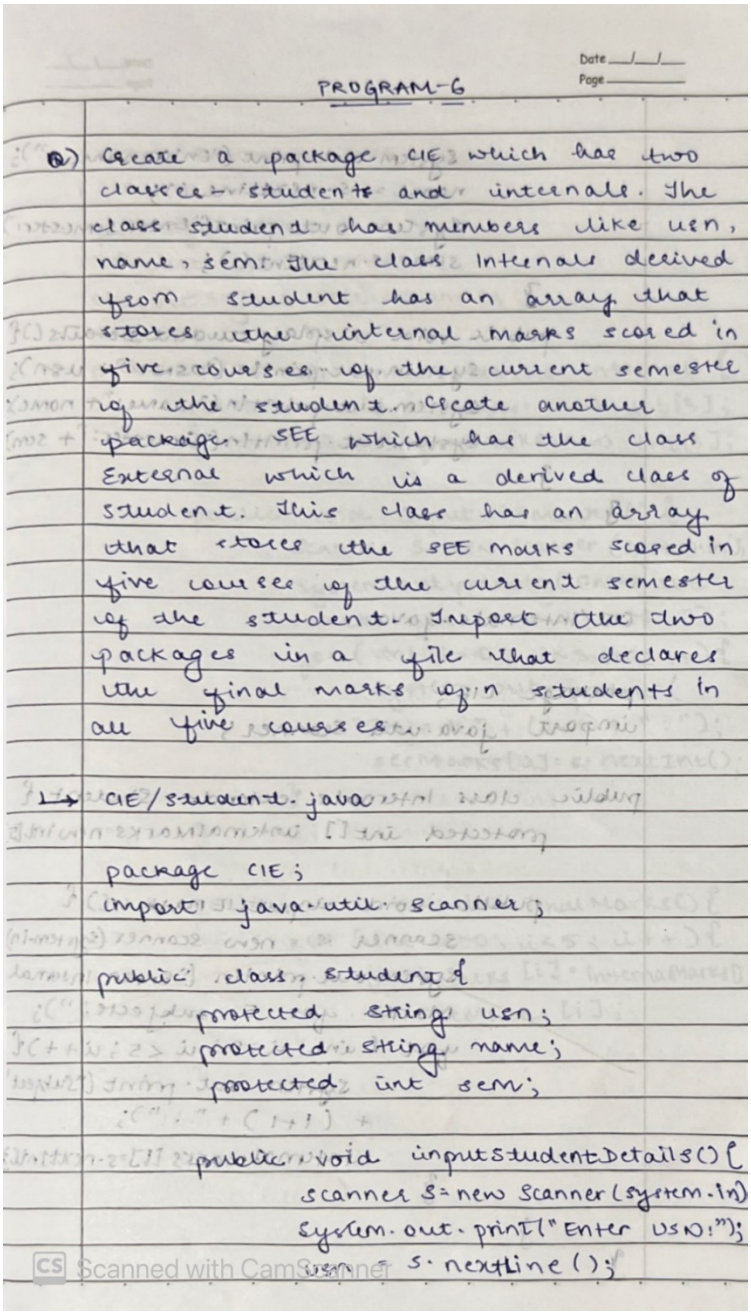

Windows PowerShell
PS C:\Users\Admin> cd D:\
PS D:\> cd 002
PS D:\002> javac Bank.java
PS D:\002> java Bank
enter customer name:
Ahmed
enter accno:
03465
enter initial balance:
15000
enter minimum balance:
10000
enter interest rate:
3
enter service charge:
2.75
Enter choice:
1.Current acc
2.Savings acc
2
Customer name is:Ahmed
Account number:3465
Aamal Magdum-1BM23CS002
account is savings type
enter choice:
1.deposit
2.withdraw
3.display balance
2
enter amount to withdraw:
3000
withdrew:3000.0 balance is:12000.0
enter choice:
1.deposit
2.withdraw
3.display balance
1
enter amount to be deposited:
26000
Deposited: 26000.0
enter choice:
1.deposit
2.withdraw
3.display balance


Windows PowerShell
Aamal Magdum-1BM23CS002
account is savings type
enter choice:
1.deposit
2.withdraw
3.display balance
2
enter amount to withdraw:
3000
withdrew:3000.0 balance is:12000.0
enter choice:
1.deposit
2.withdraw
3.display balance
1
enter amount to be deposited:
26000
Deposited: 26000.0
enter choice:
1.deposit
2.withdraw
3.display balance
3
Deposited: 1140.0
Current Balance: 39140.0
enter choice:
1.deposit
2.withdraw
3.display balance
2
enter amount to withdraw:
35000
withdrew:35000.0 balance is:4140.0
enter choice:
1.deposit
2.withdraw
3.display balance
3
Deposited: 124.2
Current Balance: 4264.2
enter choice:
1.deposit
2.withdraw
3.display balance

Program 6

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

Algorithm:



PROGRAM-6

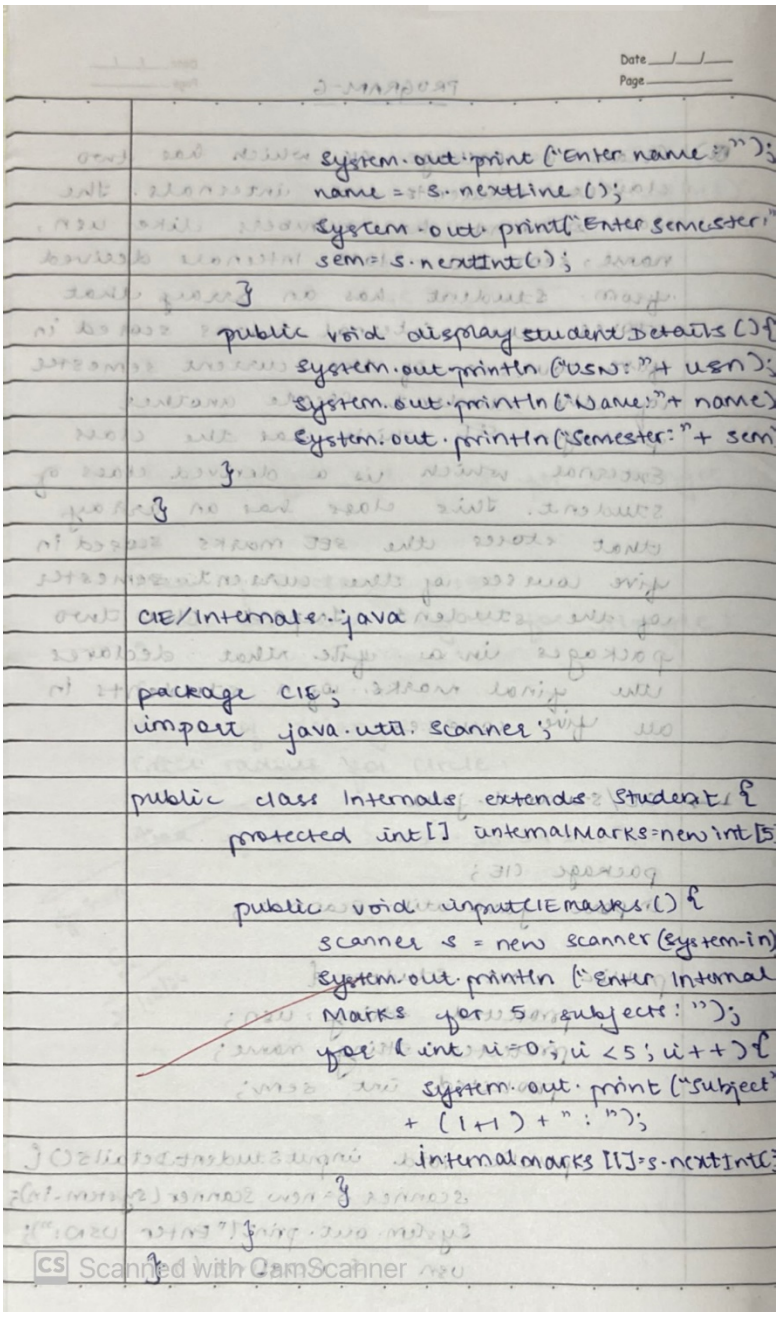
Date __/__/__
Page _____

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

1) CIE/Student.java

```
package CIE;
import java.util.Scanner;
public class Student{
    protected String usn;
    protected String name;
    protected int sem;

    public void inputStudentDetails(){
        Scanner s=new Scanner(System.in);
        System.out.print("Enter USN:");
        usn= s.nextLine();
```

Scanned with CamScanner

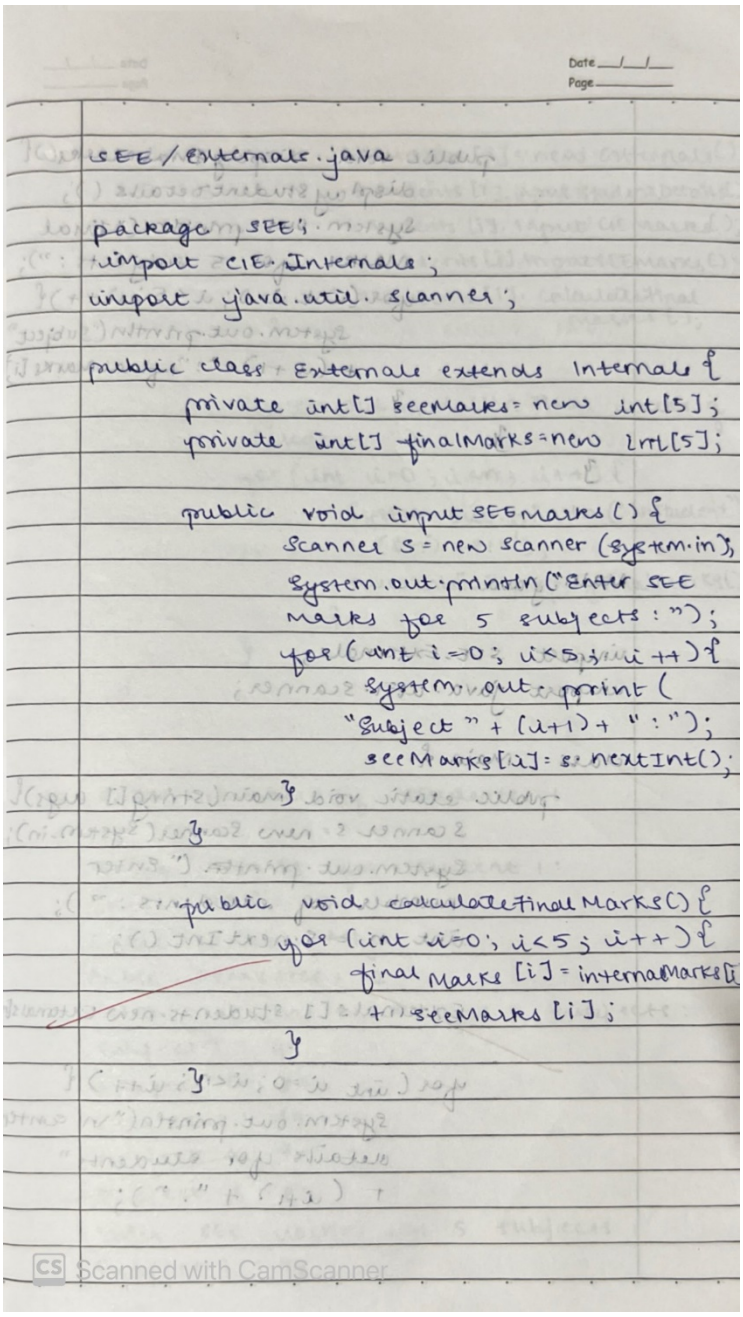


```
        System.out.print("Enter name :");
        name = s.nextLine();
        System.out.print("Enter semester:");
        sem = s.nextInt();
    }
    public void displayStudentDetails(){
        System.out.println("USN: "+ usn);
        System.out.println("Name: "+ name);
        System.out.println("Semester: "+ sem);
    }
}
```

CIE/Internals.java

```
package CIE;
import java.util.Scanner;

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

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

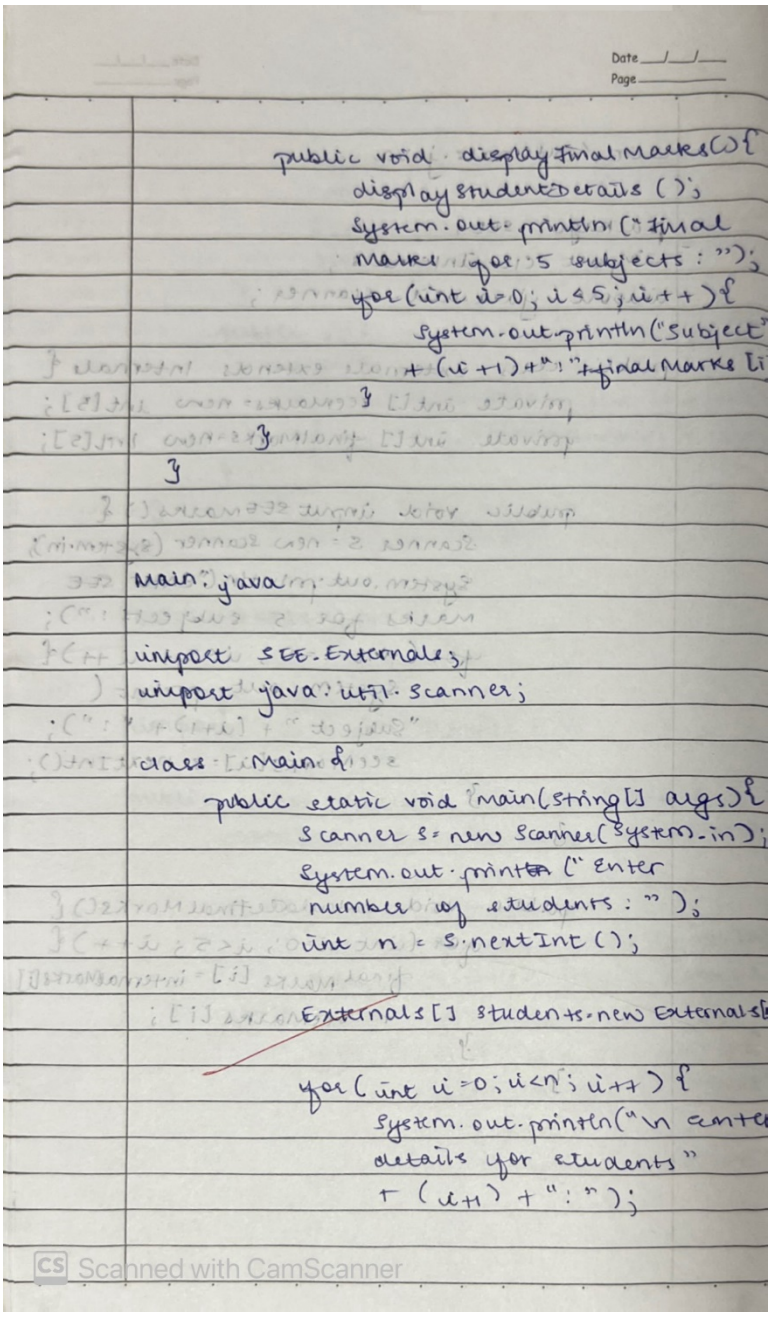
Scanned with CamScanner



SEE / External.java

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

public class External extends Internals {
    private int[] seeMarks = new int[5];
    private int[] finalMarks = new int[5];

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

    public void calculateFinalMarks() {
        for(int i=0; i<5; i++) {
            finalMarks[i] = internalMarks[i]
            + seeMarks[i];
        }
    }
```

Scanned with CamScanner

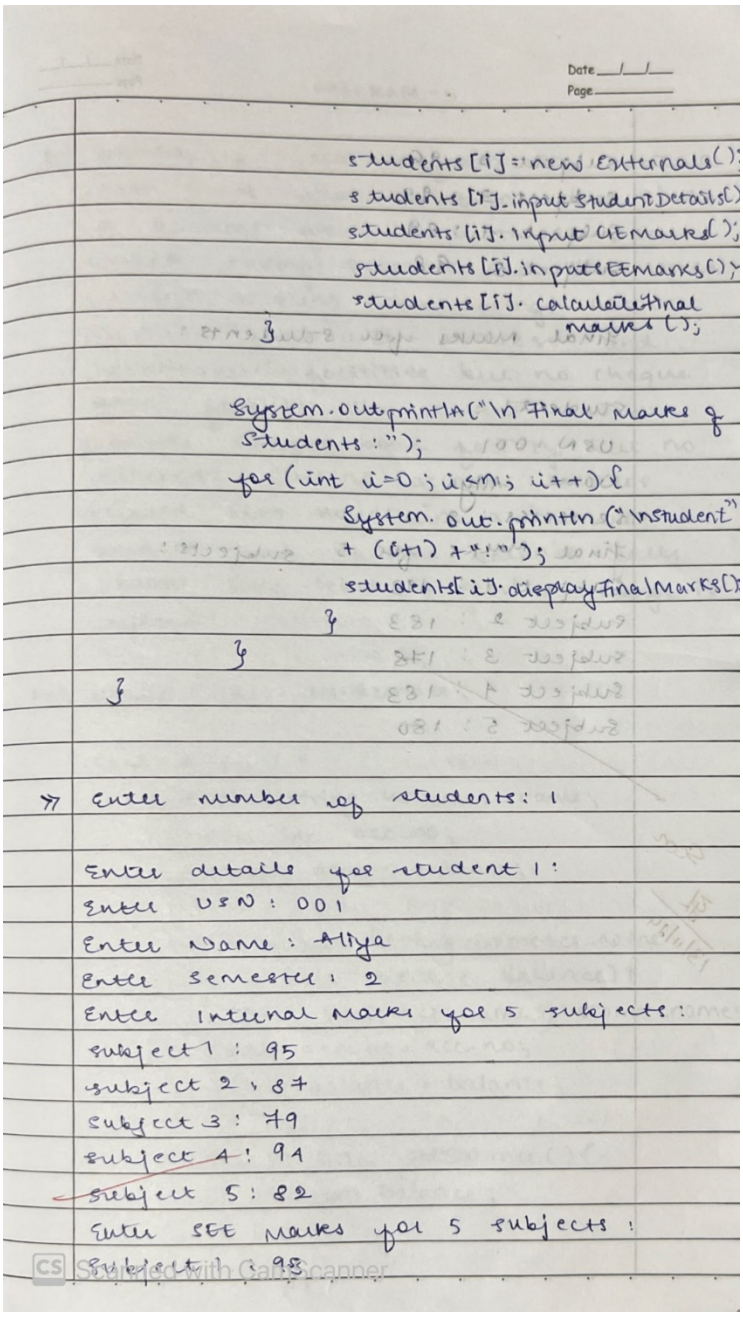


```
public void displayFinalMarks(){
    displayStudentDetails();
    System.out.println("Final
    Marks for 5 subjects :");
    for(int i=0; i<5; i++){
        System.out.println("Subject
        "+(i+1)+": "+finalMarks[i
    }
}
```

Main.java

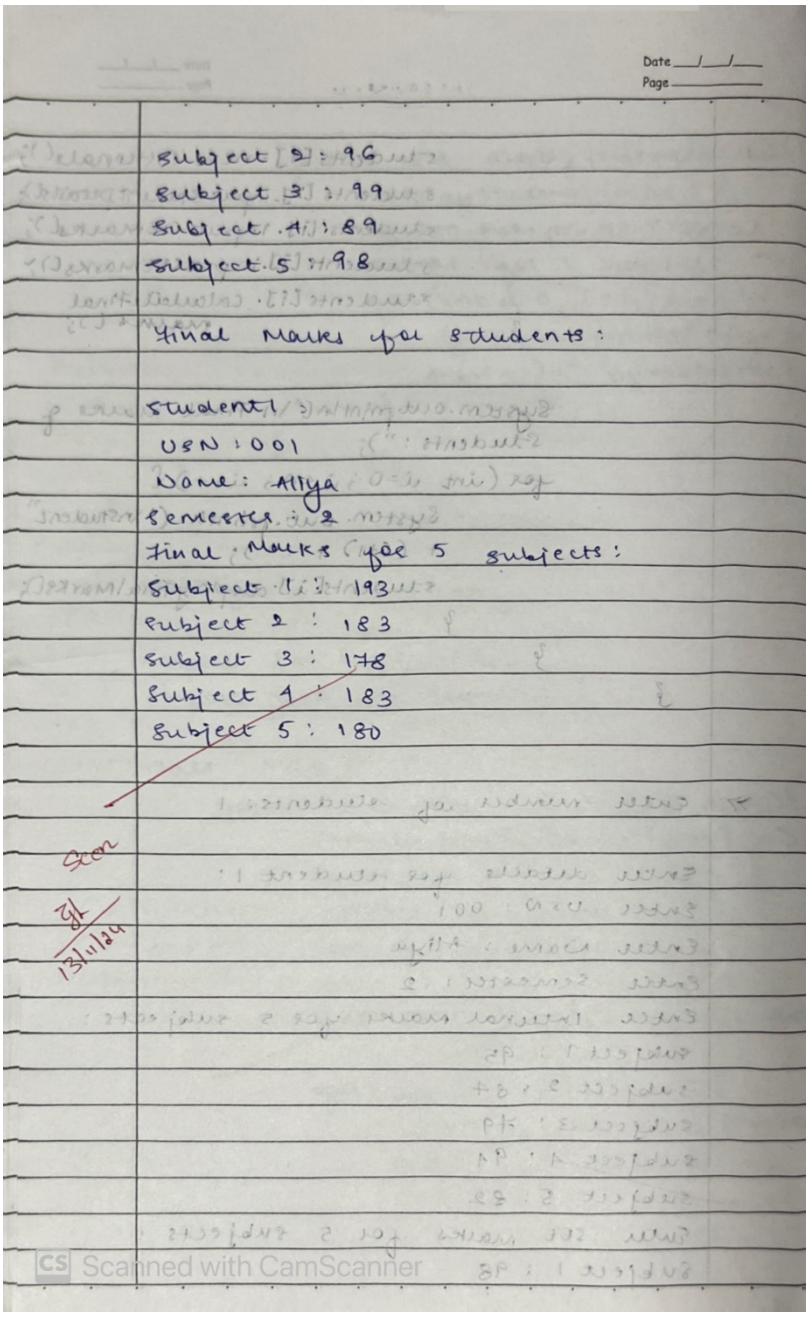
```
import SEE.External;
import java.util.Scanner;
class Main{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        System.out.print("Enter
        number of students: ");
        int n = s.nextInt();
        External[] students = new External[
        for(int i=0; i<n; i++){
            System.out.println("\n Ente
            details for students
            "+(i+1)+":");
```

CS Scanned with CamScanner



```
            students[i] = new Externals();
            students[i].inputStudentDetails();
            students[i].inputCIEMarks();
            students[i].inputSEEMarks();
            students[i].calculateFinal
                        Marks();
        }

        System.out.println("\n Final Marks of
        students :");
        for (int i=0 ; i<n; i++) {
                System.out.println("\nStudent"
                + (i+1) +":");
                students[i].displayFinalMarks();
            }
        }
    }

» Enter number of students: 1

Enter details for student 1:
Enter USN: 001
Enter Name: Aliya
Enter Semester: 2
Enter Internal Marks for 5 subjects:
subject 1: 95
subject 2: 87
subject 3: 79
subject 4: 91
subject 5: 82
Enter SEE Marks for 5 subjects:
Subject 1: 98
```

CS Scanned with CamScanner



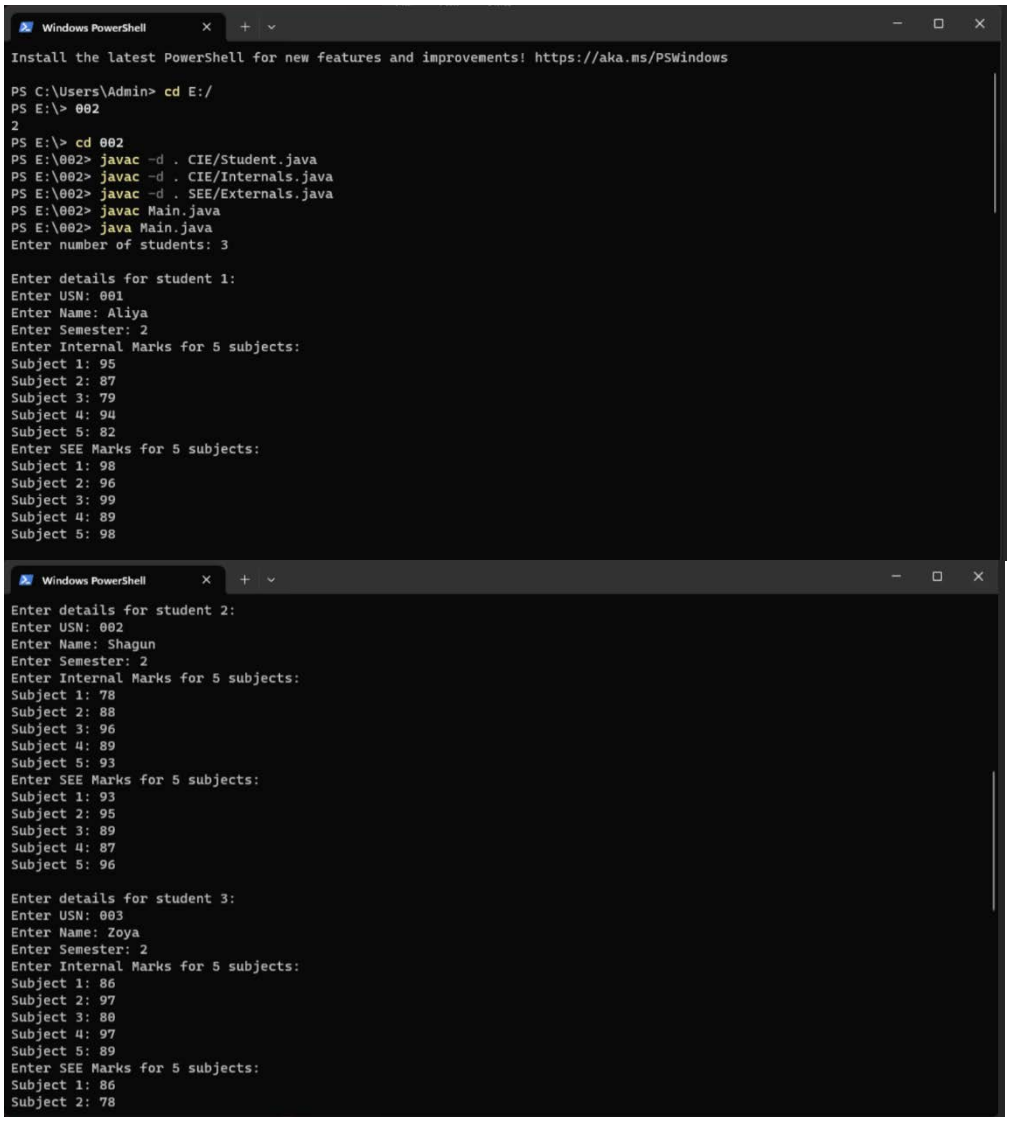
Code:

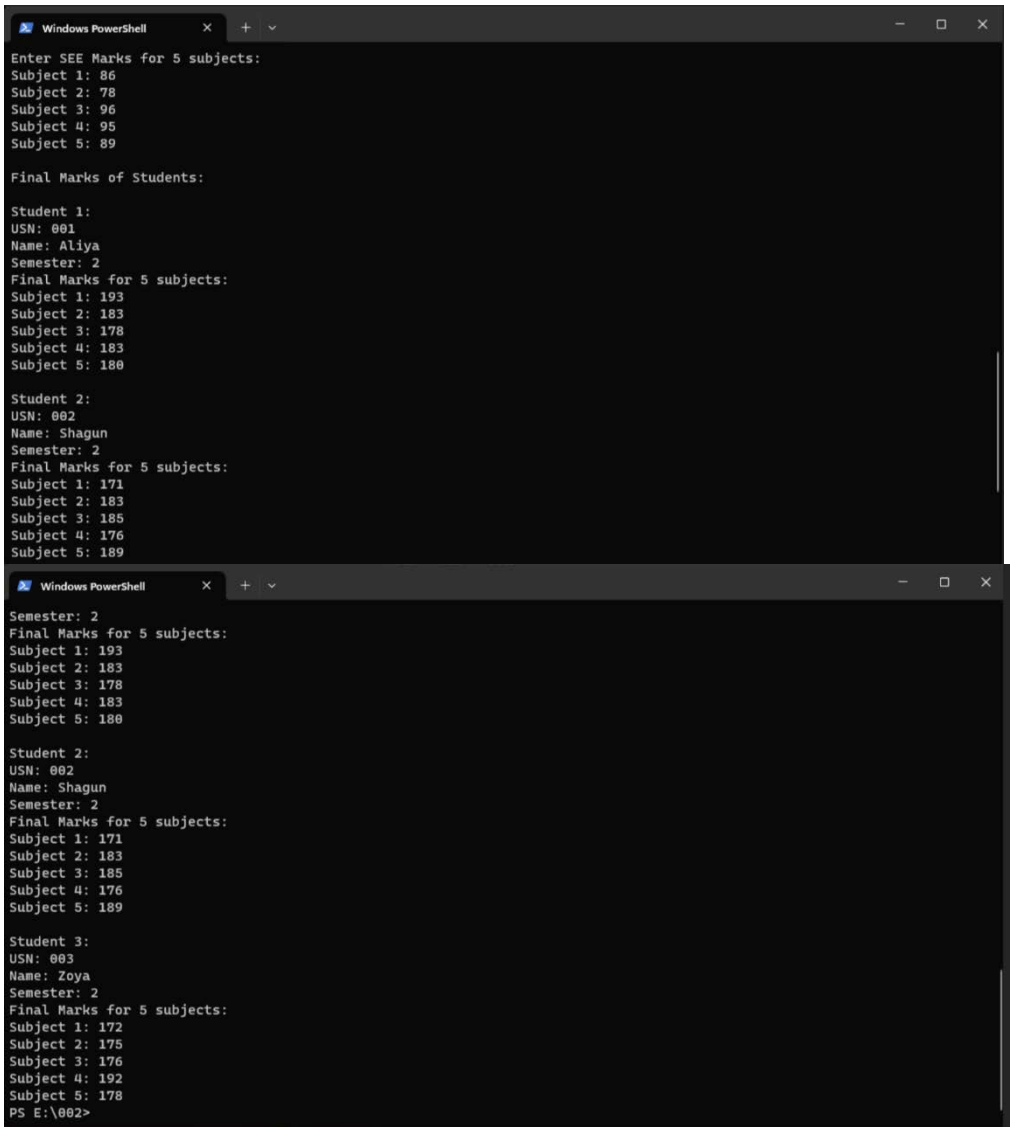
```
//CIE/Student.java
package CIE;
import java.util.Scanner;

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

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

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

//CIE/Internals.java
package CIE;
import java.util.Scanner;

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

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

//SEE/Externals.java
package SEE;
import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {
    private int[] seeMarks = new int[5];
    private int[] finalMarks = new int[5];

    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE Marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
```

```
            System.out.print("Subject " + (i + 1) + ": ");
            seeMarks[i] = s.nextInt();
        }
    }

    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = internalMarks[i] + seeMarks[i];
        }
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        System.out.println("Final Marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
        }
    }
}

//Main.java
import SEE.Externals;
import java.util.Scanner;

class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter number of students: ");
        int n = s.nextInt();

        Externals[] students = new Externals[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for student " + (i + 1) + ":");
            students[i] = new Externals();
            students[i].inputStudentDetails();
            students[i].inputCIEmarks();
            students[i].inputSEEmarks();
            students[i].calculateFinalMarks();
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nStudent " + (i + 1) + ":");
            students[i].displayFinalMarks();
        }
    }
}
```

Output:

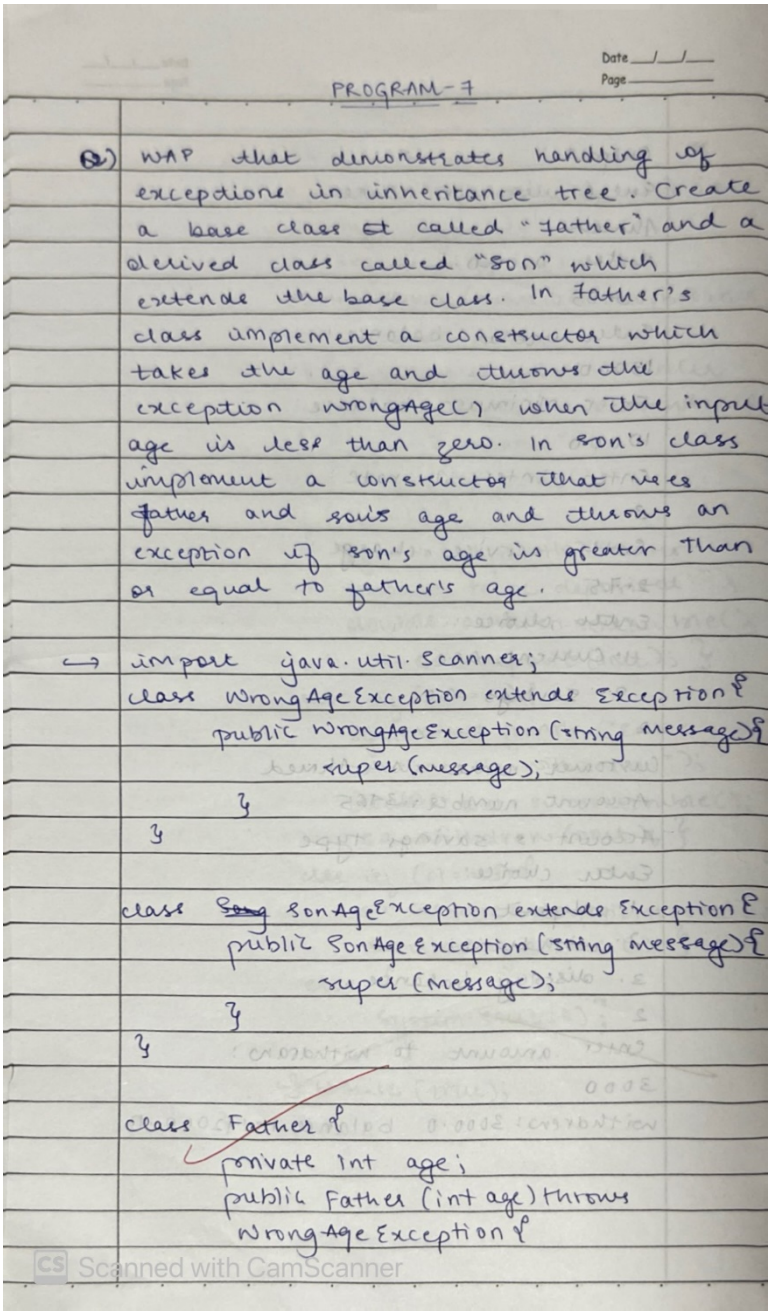

Windows PowerShell
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\Users\Admin> cd E:/
PS E:\> 002
2
PS E:\> cd 002
PS E:\002> javac -d . CIE/Student.java
PS E:\002> javac -d . CIE/Internals.java
PS E:\002> javac -d . SEE/Externals.java
PS E:\002> javac Main.java
PS E:\002> java Main.java
Enter number of students: 3
Enter details for student 1:
Enter USN: 001
Enter Name: Aliya
Enter Semester: 2
Enter Internal Marks for 5 subjects:
Subject 1: 95
Subject 2: 87
Subject 3: 79
Subject 4: 94
Subject 5: 82
Enter SEE Marks for 5 subjects:
Subject 1: 98
Subject 2: 96
Subject 3: 99
Subject 4: 89
Subject 5: 98
Windows PowerShell
Enter details for student 2:
Enter USN: 002
Enter Name: Shagun
Enter Semester: 2
Enter Internal Marks for 5 subjects:
Subject 1: 78
Subject 2: 88
Subject 3: 96
Subject 4: 89
Subject 5: 93
Enter SEE Marks for 5 subjects:
Subject 1: 93
Subject 2: 95
Subject 3: 89
Subject 4: 87
Subject 5: 96
Enter details for student 3:
Enter USN: 003
Enter Name: Zoya
Enter Semester: 2
Enter Internal Marks for 5 subjects:
Subject 1: 86
Subject 2: 97
Subject 3: 80
Subject 4: 97
Subject 5: 89
Enter SEE Marks for 5 subjects:
Subject 1: 86
Subject 2: 78


Windows PowerShell
Enter SEE Marks for 5 subjects:
Subject 1: 86
Subject 2: 78
Subject 3: 96
Subject 4: 95
Subject 5: 89
Final Marks of Students:
Student 1:
USN: 001
Name: Aliya
Semester: 2
Final Marks for 5 subjects:
Subject 1: 193
Subject 2: 183
Subject 3: 178
Subject 4: 183
Subject 5: 180
Student 2:
USN: 002
Name: Shagun
Semester: 2
Final Marks for 5 subjects:
Subject 1: 171
Subject 2: 183
Subject 3: 185
Subject 4: 176
Subject 5: 189
Windows PowerShell
Semester: 2
Final Marks for 5 subjects:
Subject 1: 193
Subject 2: 183
Subject 3: 178
Subject 4: 183
Subject 5: 180
Student 2:
USN: 002
Name: Shagun
Semester: 2
Final Marks for 5 subjects:
Subject 1: 171
Subject 2: 183
Subject 3: 185
Subject 4: 176
Subject 5: 189
Student 3:
USN: 003
Name: Zoya
Semester: 2
Final Marks for 5 subjects:
Subject 1: 172
Subject 2: 175
Subject 3: 176
Subject 4: 192
Subject 5: 178
PS E:\002>

Program 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called ‘Father’ and a derived class called ‘Son’ which extends the base class. In Father’s class implement a constructor which takes the age and throws the exception wrongAge() when the input age is less than zero. In Son’s class implement a constructor that uses father and son’s age and throws an exception if son’s age is greater than or equal to father’s age.

Algorithm:

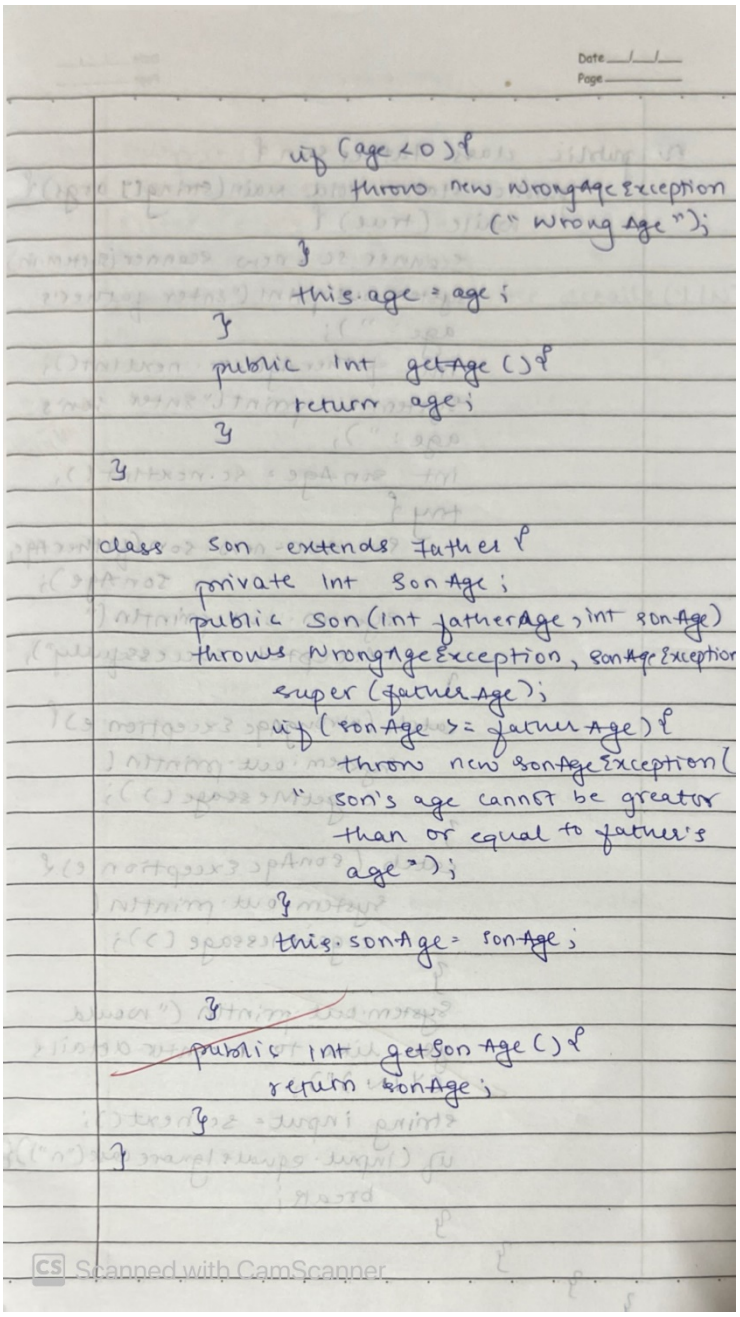


PROGRAM-7

Q) WAP that demonstrates handling of exceptions in inheritance tree. Create a base class called "father" and a derived class called "son" which extends the base class. In father's class implement a constructor which takes the age and throws the exception wrongAge() when the input age is less than zero. In son's class implement a constructor that uses father and sons age and throws an exception if son's age is greater than or equal to father's age.

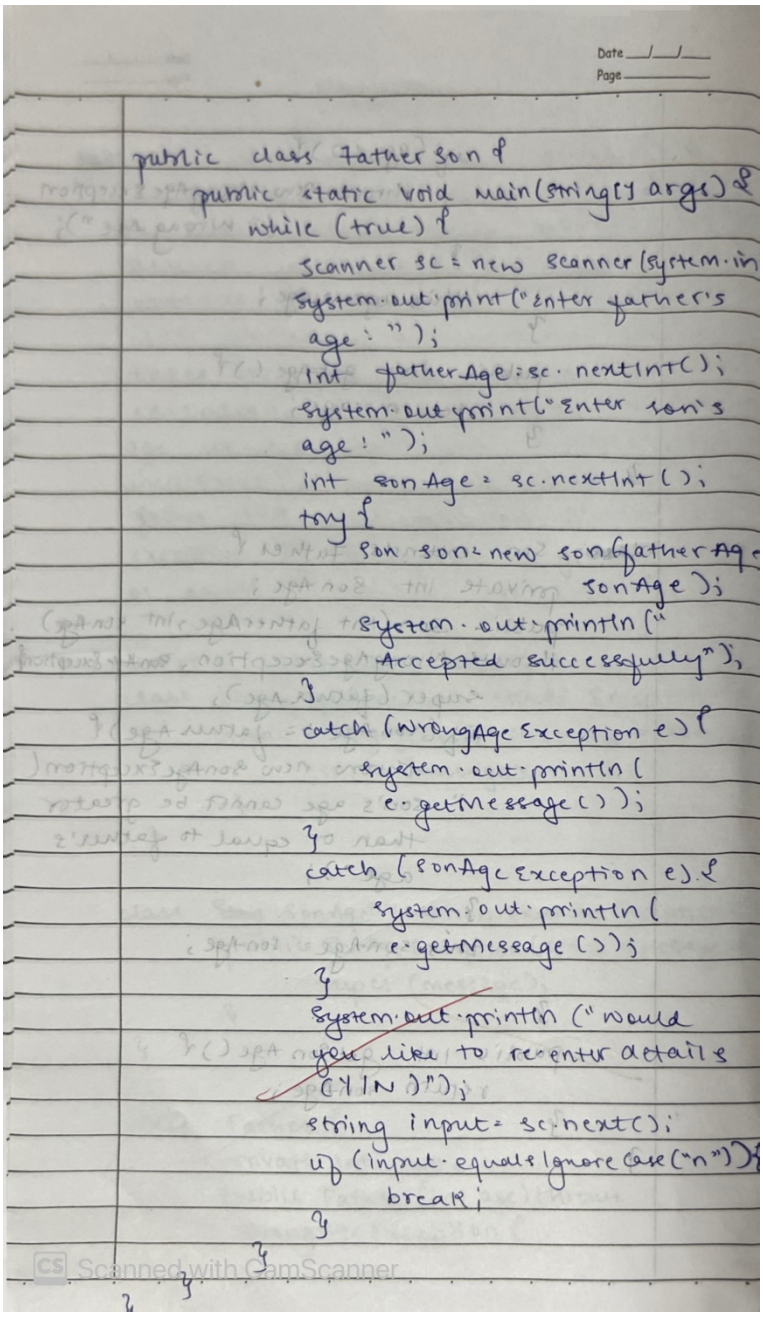
```
import java.util.Scanner;
class WrongAgeException extends Exception{
    public WrongAgeException(String message){
        super(message);
    }
}

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

class Father{
    private int age;
    public Father(int age) throws
    WrongAgeException{
```

Scanned with CamScanner



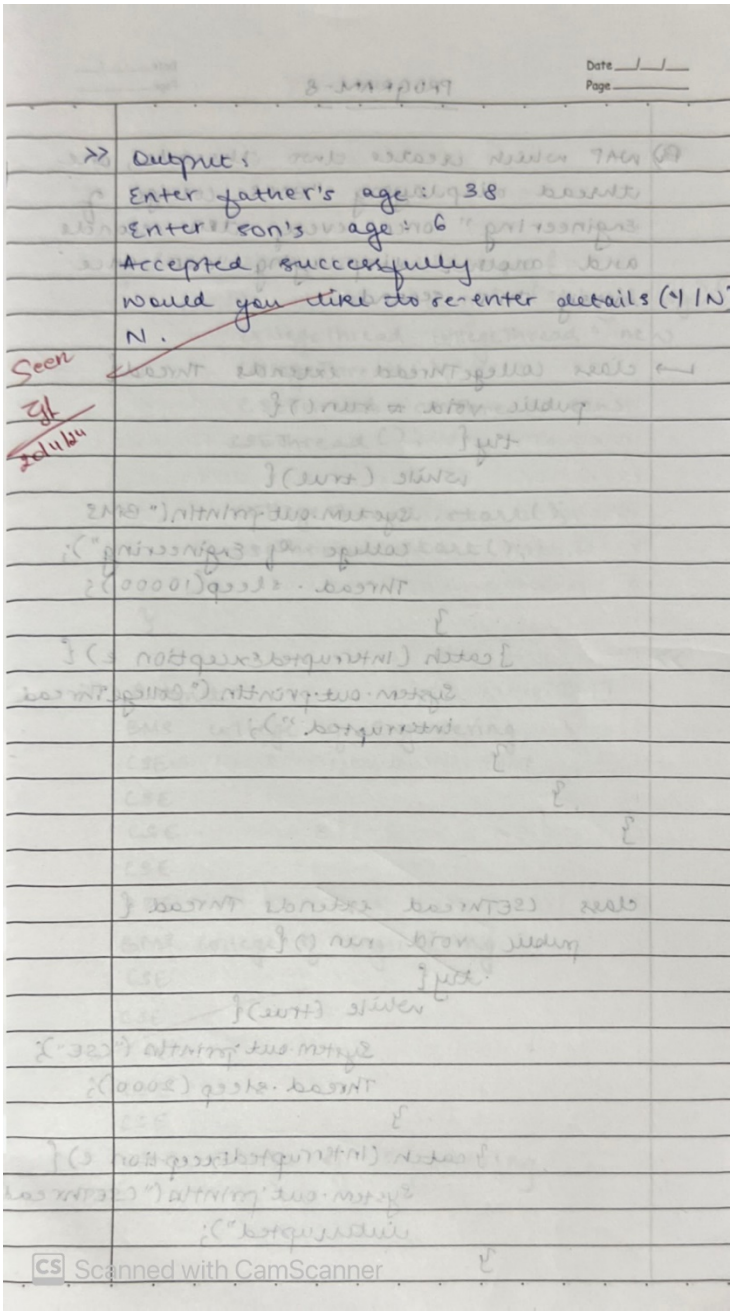
```
        if (age < 0) {
            throw new WrongAgeException
                ("Wrong Age");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

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;
    }
}
```

CS Scanned with CamScanner



```
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 successfully");
            }
            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;
            }
        }
    }
}
```

Scanned with CamScanner



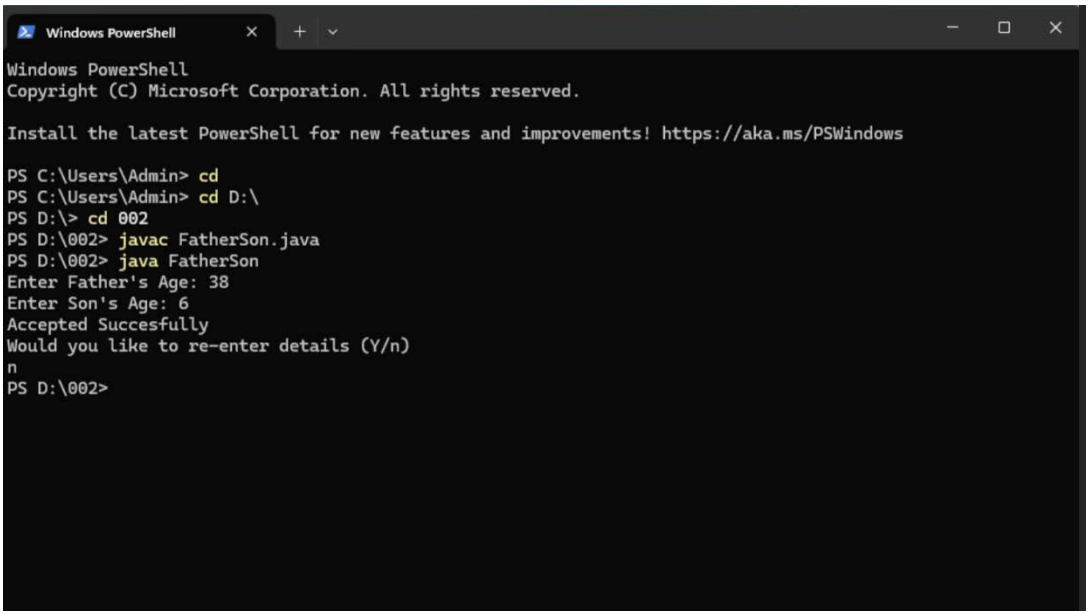
Code:

```
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 successfully");
            }
            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;
        }
    }
  }
}
```

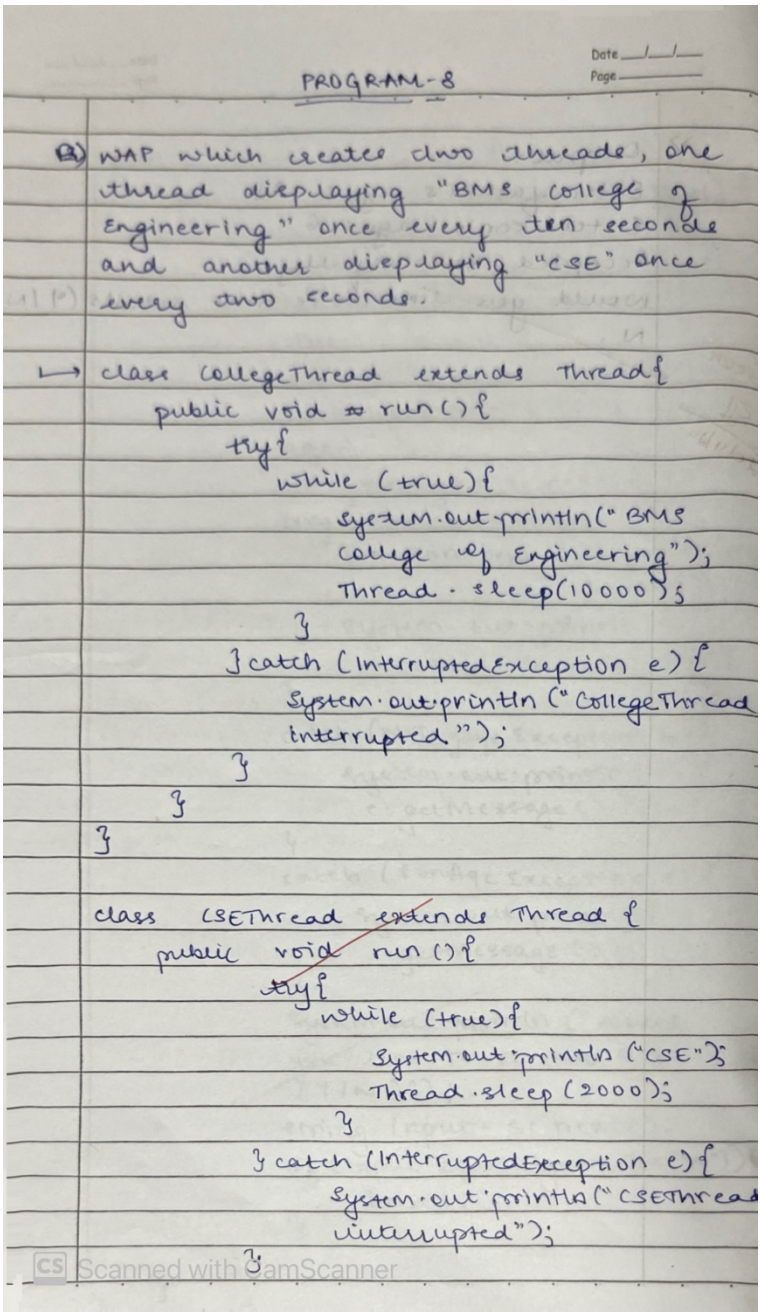
Output:



Program 8

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

Algorithm:



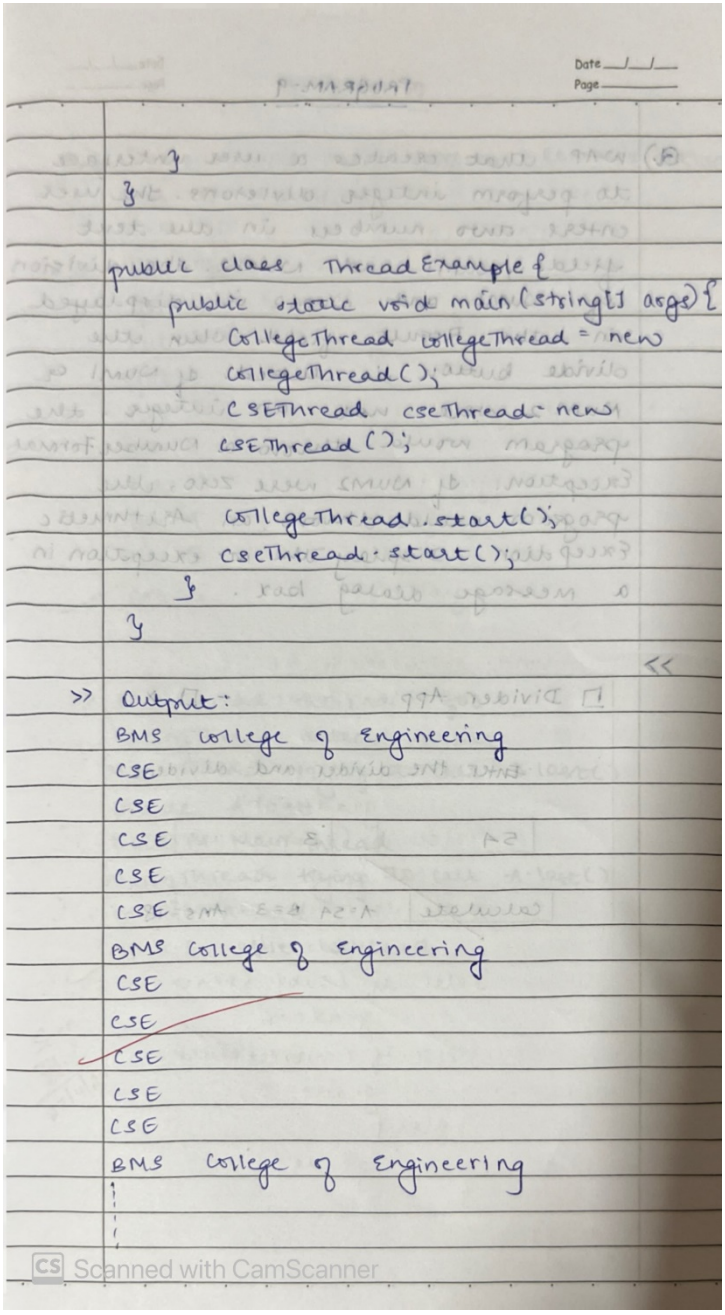
PROGRAM-8

Date __/__/__
Page

8) WAP which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

```
class CollegeThread extends Thread{
    public void run(){
        try{
            while (true){
                System.out.println("BMS
                College of Engineering");
                Thread.sleep(10000);
            }
        }catch (InterruptedException e){
            System.out.println ("CollegeThread
            interrupted");
        }
    }
}

class CSEThread extends Thread {
    public void run (){
        try{
            while (true){
                System.out.println ("CSE");
                Thread.sleep (2000);
            }
        } catch (InterruptedException e){
            System.out.println("CSEThread
            interrupted");
        }
```

Scanned with CamScanner

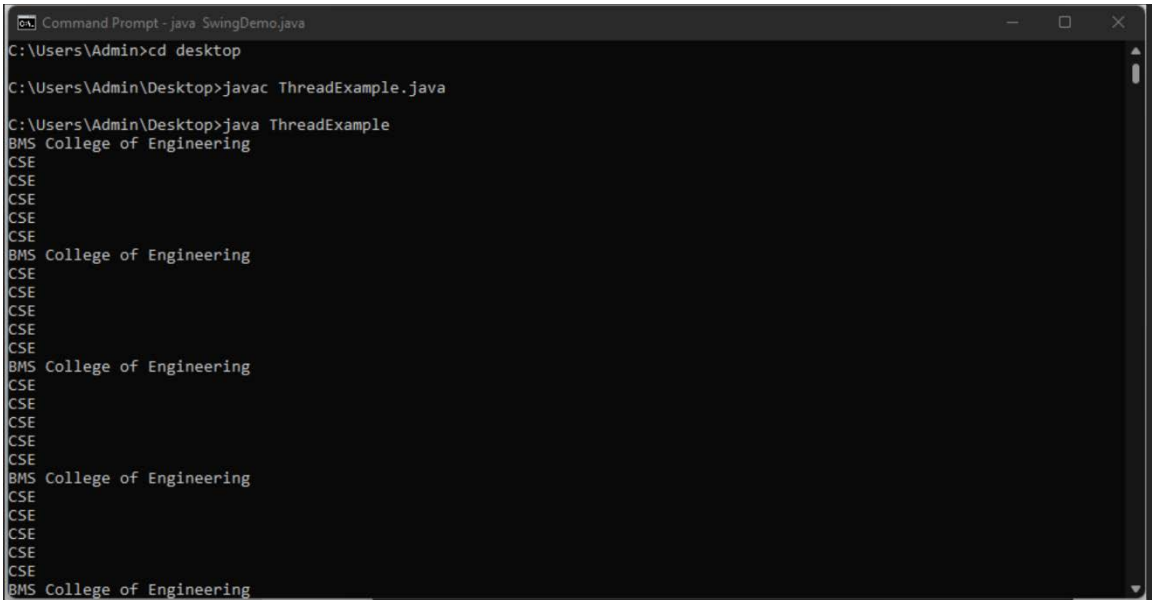


Code:

```
class CollegeThread extends Thread {
   public void run() {
      try {
         while (true) {
            System.out.println("BMS College of Engineering");
            Thread.sleep(10000);
         }
      } catch (InterruptedException e) {
```

```
            System.out.println("CollegeThread interrupted");
        }
    }
}

class CSEThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("CSEThread interrupted");
        }
    }
}

public class ThreadExample {
    public static void main(String[] args) {
        CollegeThread collegeThread = new CollegeThread();
        CSEThread cseThread = new CSEThread();

        collegeThread.start();
        cseThread.start();
    }
}
```

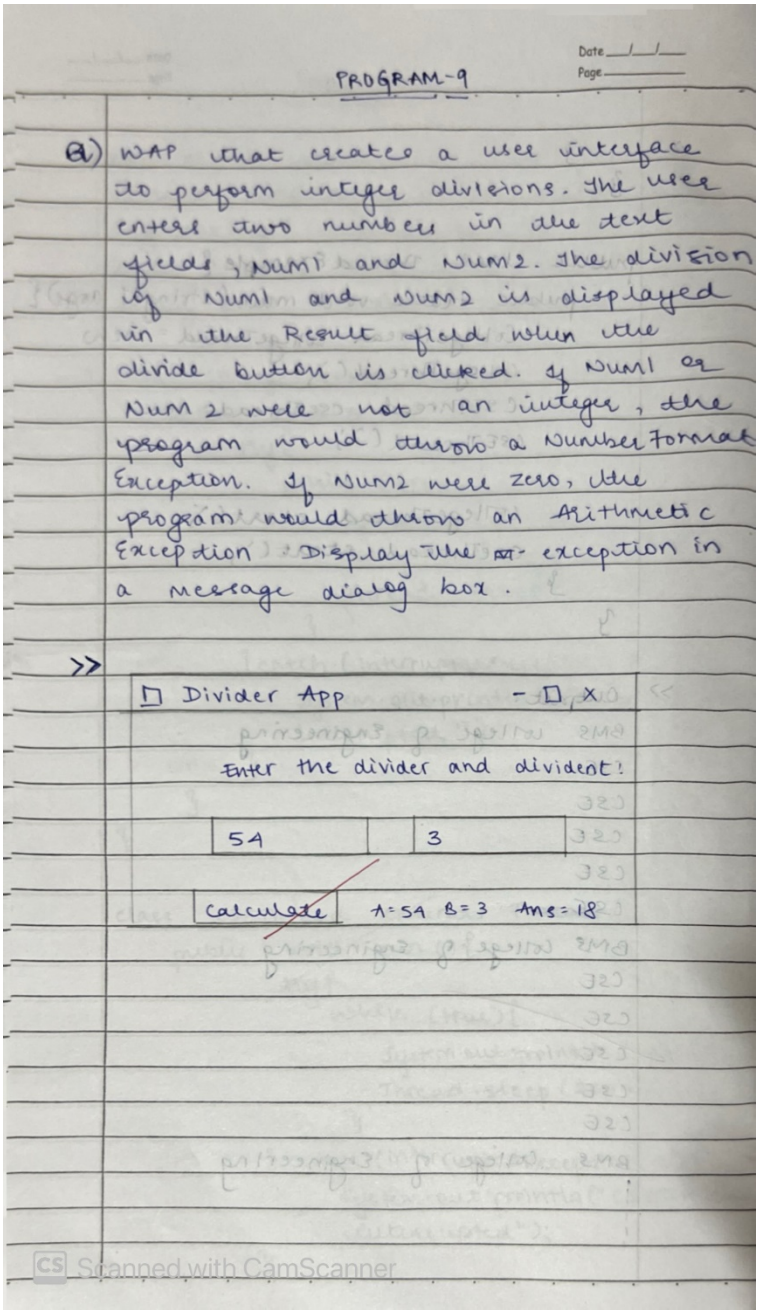
Output:



Program 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

Algorithm:

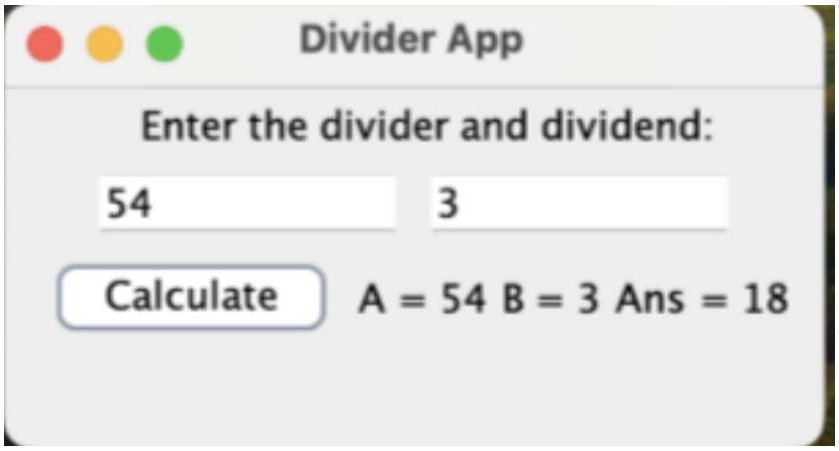


Code:

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

class SwingDemo { SwingDemo() {

    JFrame jfrm = new JFrame("Divider App");
    jfrm.setSize(275, 150);
    jfrm.setLayout(new FlowLayout());

    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JLabel jlab = new JLabel("Enter the divider and dividend:");
    JTextField ajtf = new JTextField(8);
    JTextField bjtf = new JTextField(8);
    JButton button = new JButton("Calculate");
    JLabel err = new JLabel();
    JLabel alab = new JLabel();
    JLabel blab = new JLabel();
    JLabel anslab = new JLabel();

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

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

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

             alab.setText("A = " + a);
             blab.setText("B = " + b);
             anslab.setText("Ans = " + ans);
```

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

   jfrm.setVisible(true);
}

public static void main(String args[]) {

   SwingUtilities.invokeLater(new Runnable() {
      public void run() {
         new SwingDemo();
      }
   });
}


}
```

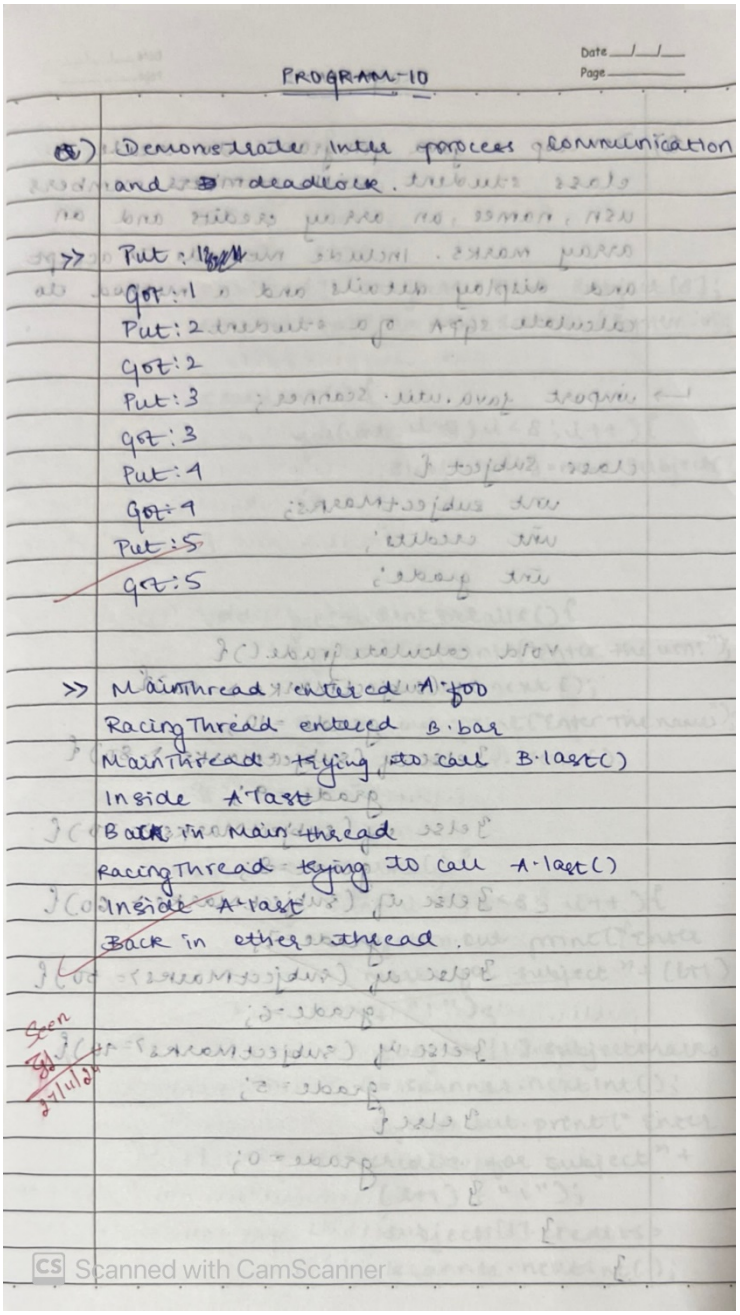
Output:


Divider App
Enter the divider and dividend:
54
3
Calculate
A = 54 B = 3 Ans = 18

Program 10

Demonstrate Inter process communication and deadlock.

Algorithm:



Code:

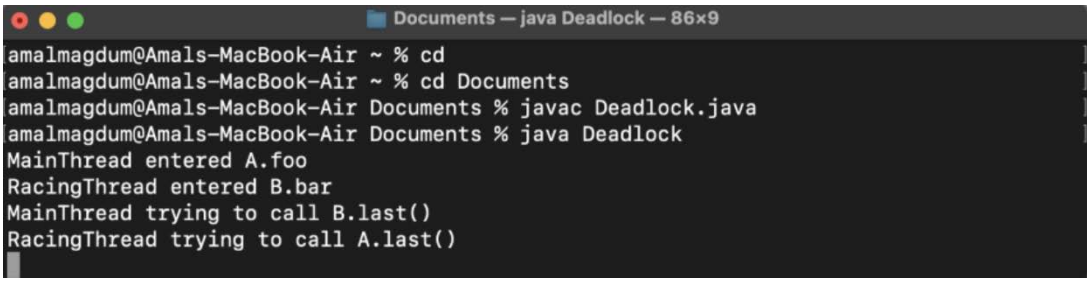
//Deadlock

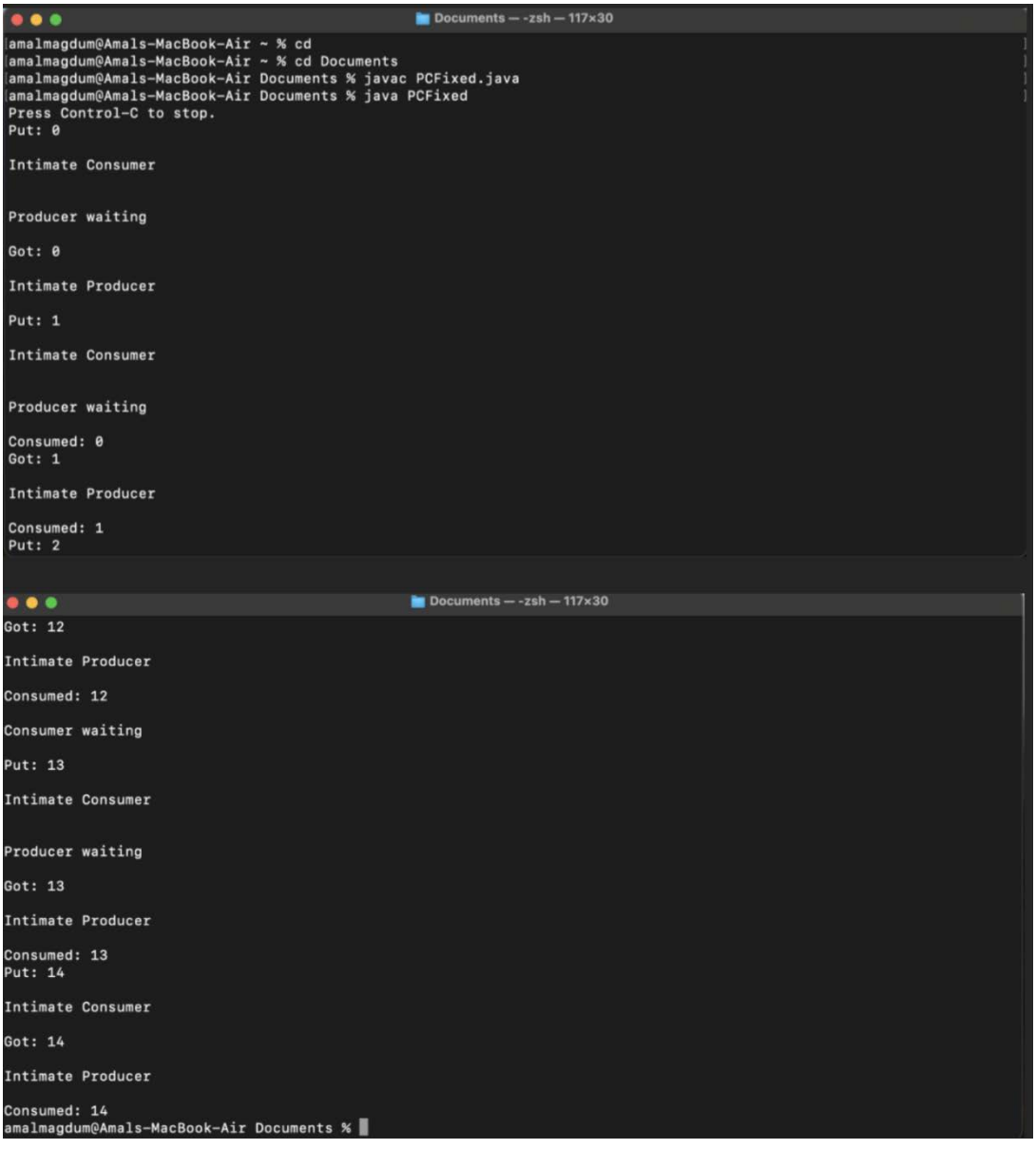
```
class A {
   synchronized void foo(B b) {
      String name = Thread.currentThread().getName();
      System.out.println(name + " entered A.foo");

      try {
         Thread.sleep(1000);
      } catch (Exception e) {
         System.out.println("A Interrupted");
      }

      System.out.println(name + " trying to call B.last()");
      b.last();
   }

   synchronized void last() {
      System.out.println("Inside A.last");
   }
}

class B {
   synchronized void bar(A a) {
      String name = Thread.currentThread().getName();
      System.out.println(name + " entered B.bar");

      try {
         Thread.sleep(1000);
      } catch (Exception e) {
         System.out.println("B Interrupted");
      }

      System.out.println(name + " trying to call A.last()");
      a.last();
   }

   synchronized void last() {
      System.out.println("Inside B.last");
   }
}

class Deadlock implements Runnable {
   A a = new A();
   B b = new B();

   Deadlock() {
      Thread.currentThread().setName("MainThread");
      Thread t = new Thread(this, "RacingThread");
      t.start();
```

```
        a.foo(b);
        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }

    public static void main(String args[]) {
        new Deadlock();
    }
}


//PCFixed
class Q {
    int n;
    boolean valueSet = false;

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

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

```
        System.out.println("Put: " + n);
        System.out.println("\nIntimate Consumer\n");
        notify();
    }
}

class Producer implements Runnable {
    Q q;

    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }

    public void run() {
        int i = 0;
        while (i < 15) {
            q.put(i++);
        }
    }
}

class Consumer implements Runnable {
    Q q;

    Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }

    public void run() {
        int i = 0;
        while (i < 15) {
            int r = q.get();
            System.out.println("Consumed: " + r);
            i++;
        }
    }
}

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

Output:


Documents — java Deadlock — 86×9
amalmagdum@Amals-MacBook-Air ~ % cd
amalmagdum@Amals-MacBook-Air ~ % cd Documents
amalmagdum@Amals-MacBook-Air Documents % javac Deadlock.java
amalmagdum@Amals-MacBook-Air Documents % java Deadlock
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
RacingThread trying to call A.last()


Documents — -zsh — 117×30
amalmagdum@Amals-MacBook-Air ~ % cd
amalmagdum@Amals-MacBook-Air ~ % cd Documents
amalmagdum@Amals-MacBook-Air Documents % javac PCFixed.java
amalmagdum@Amals-MacBook-Air Documents % java PCFixed
Press Control-C to stop.
Put: 0
Intimate Consumer
Producer waiting
Got: 0
Intimate Producer
Put: 1
Intimate Consumer
Producer waiting
Consumed: 0
Got: 1
Intimate Producer
Consumed: 1
Put: 2
Documents — -zsh — 117×30
Got: 12
Intimate Producer
Consumed: 12
Consumer waiting
Put: 13
Intimate Consumer
Producer waiting
Got: 13
Intimate Producer
Consumed: 13
Put: 14
Intimate Consumer
Got: 14
Intimate Producer
Consumed: 14
amalmagdum@Amals-MacBook-Air Documents %