# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

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

**LAB REPORT**
**on**

# Object Oriented Java Programming

# (23CS3PCOOJ)

***Submitted by***

**Utkrisht Umang (1BM23CS355)**

***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 **Utkrisht Umang (1BM23CS355),** 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.

| | |
|---|---|
| Ms. Ambuja K<br>Assistant Professor<br>Department of CSE, BMSCE | Dr. Kavitha Sooda<br>Professor & HOD<br>Department of CSE, BMSCE |

# Index

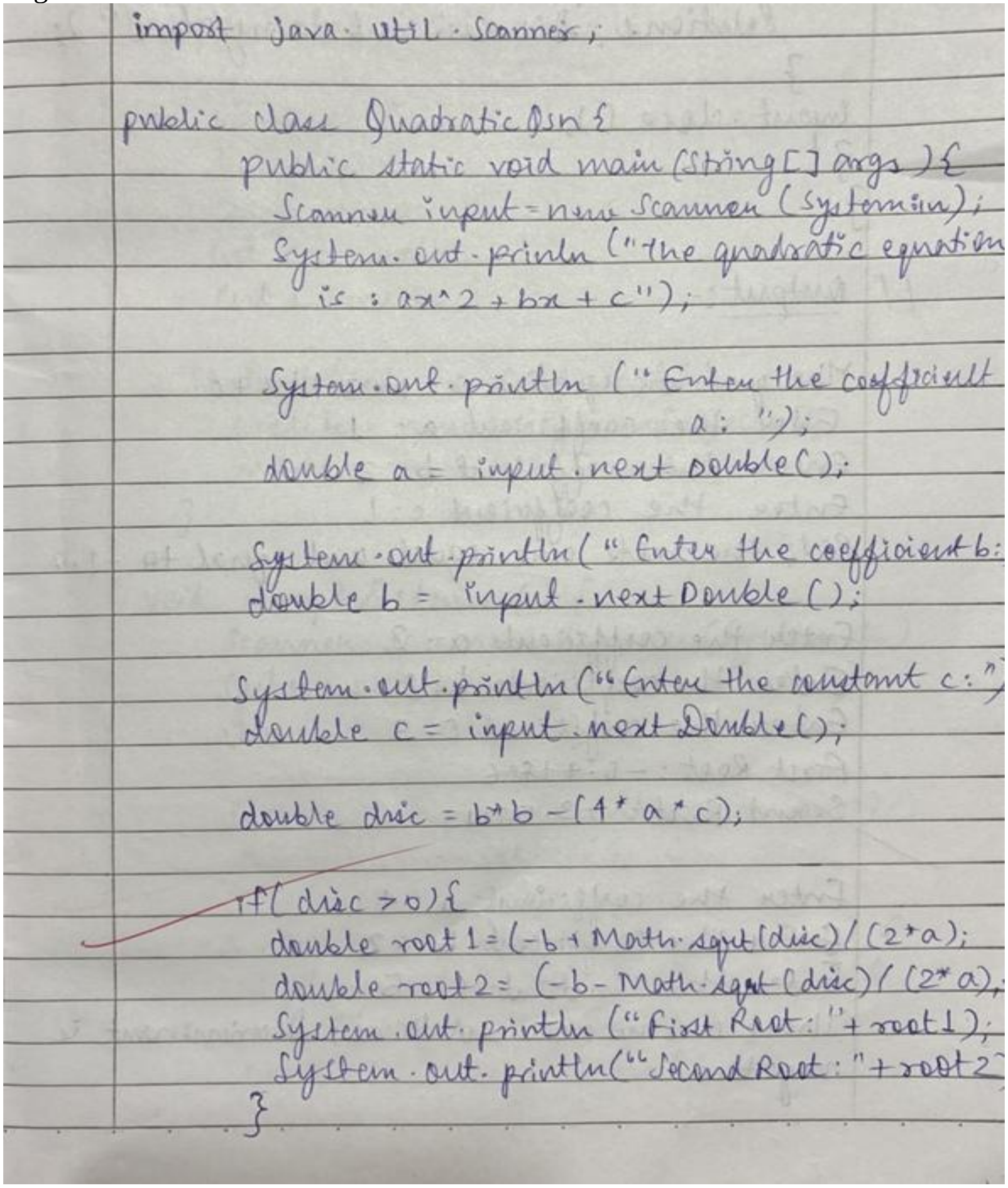
Github Link:
https://github.com/utk1college/OOJ_LAB

**Program 1**

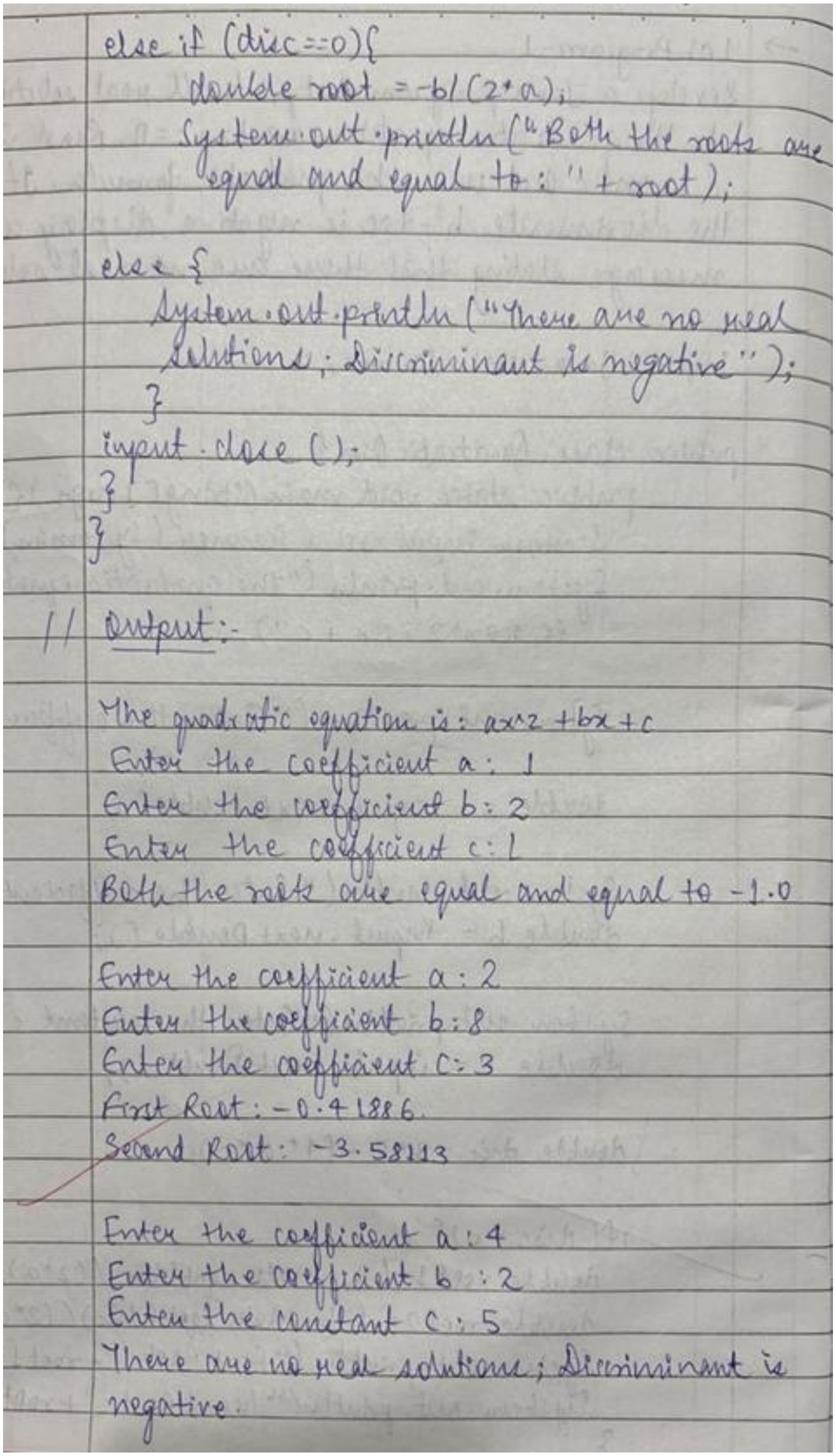
Implement Quadratic Equation

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:



```
import Java.util.Scanner;

public class QuadraticQsn{
    public static void main(String[] args){
     Scanner input=new Scanner(System.in);
     System.out.println("the quadratic equation
        is : ax^2 + bx + c");

     System.out.println("Enter the coefficient
                          a: ");
     double a = input.nextDouble();

     System.out.println("Enter the coefficient b:
     double b = input.nextDouble();

    System.out.println("Enter the constant c:")
    double c = input.nextDouble();

    double disc = b*b - (4*a*c);

    if( disc >0){
     double root1=(-b+Math.sqrt(disc)/(2*a);
     double root2= (-b-Math.sqrt(disc)/(2*a);
     System.out.println("First Root: "+root1);
     System.out.println("Second Root: "+root2
    }
```



```
else if (disc==0){
    double root = -b/(2*a);
    System.out.println("Both the roots are
     equal and equal to: " + root);

else {
   System.out.println("There are no real
    solutions; Discriminant is negative");
   }
input.close();
}
}
```

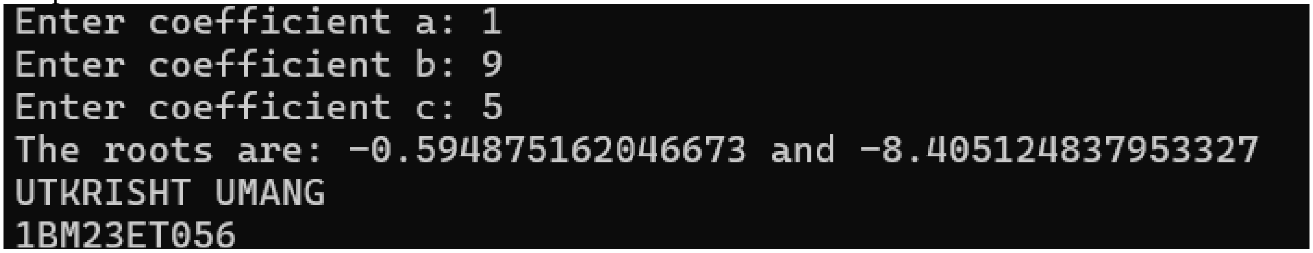
// Output:-

```
The quadratic equation is: ax^2 +bx +c
Enter the coefficient a: 1
Enter the coefficient b: 2
Enter the coefficient c: 1
Both the roots are equal and equal to -1.0

Enter the coefficient a: 2
Enter the coefficient b: 8
Enter the coefficient c: 3
First Root: -0.41886.
Second Root: -3.58113

Enter the coefficient a: 4
Enter the coefficient b: 2
Enter the constant c: 5
There are no real solutions; Discriminant is
negative.
```

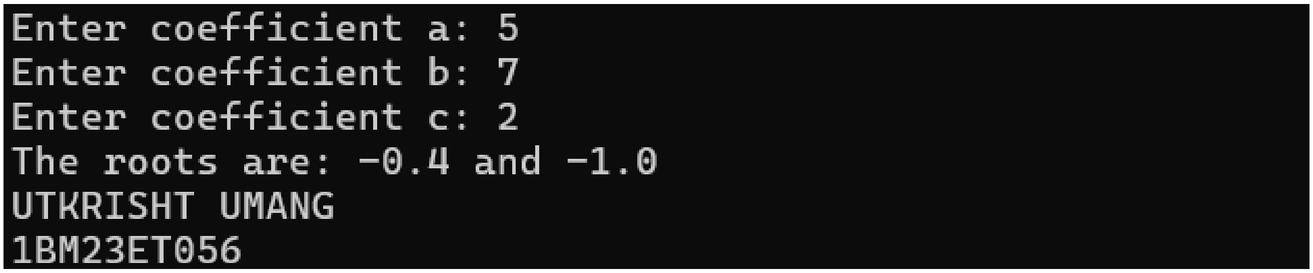
Code:

```
import java.util.Scanner;

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

        System.out.print("Enter coefficient a: ");
        double a = sc.nextDouble();
        System.out.print("Enter coefficient b: ");
        double b = sc.nextDouble();
        System.out.print("Enter coefficient c: ");
        double c = sc.nextDouble();

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

        if (disc > 0) {
            double root1 = (-b + Math.sqrt(disc)) / (2 * a);
            double root2 = (-b - Math.sqrt(disc)) / (2 * a);
            System.out.println("The roots are: " + root1 + " and " + root2);
        } else if (disc == 0) {
            double root = -b / (2 * a);
            System.out.println("The root is: " + root);
        } else {
            System.out.println("There are no real solutions; Discriminant is negative");
        }

        System.out.println("UTKRISHT UMANG\n1BM23ET056");
        sc.close();
    }
}
```

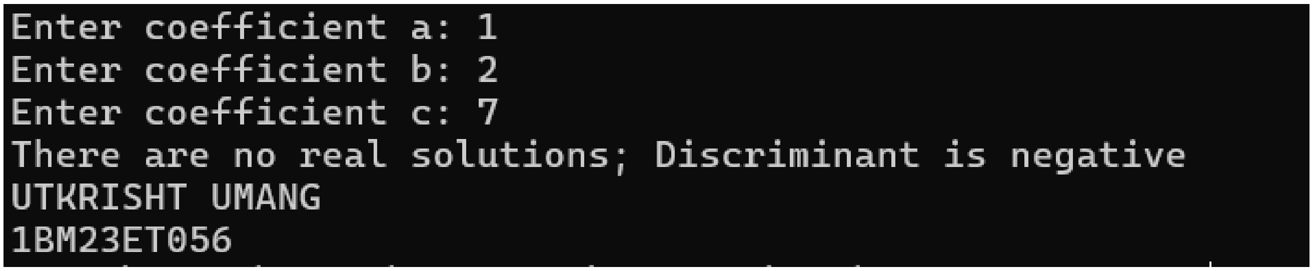
Output



```
Enter coefficient a: 1
Enter coefficient b: 9
Enter coefficient c: 5
The roots are: -0.594875162046673 and -8.405124837953327
UTKRISHT UMANG
1BM23ET056
```



```
Enter coefficient a: 5
Enter coefficient b: 7
Enter coefficient c: 2
The roots are: -0.4 and -1.0
UTKRISHT UMANG
1BM23ET056
```

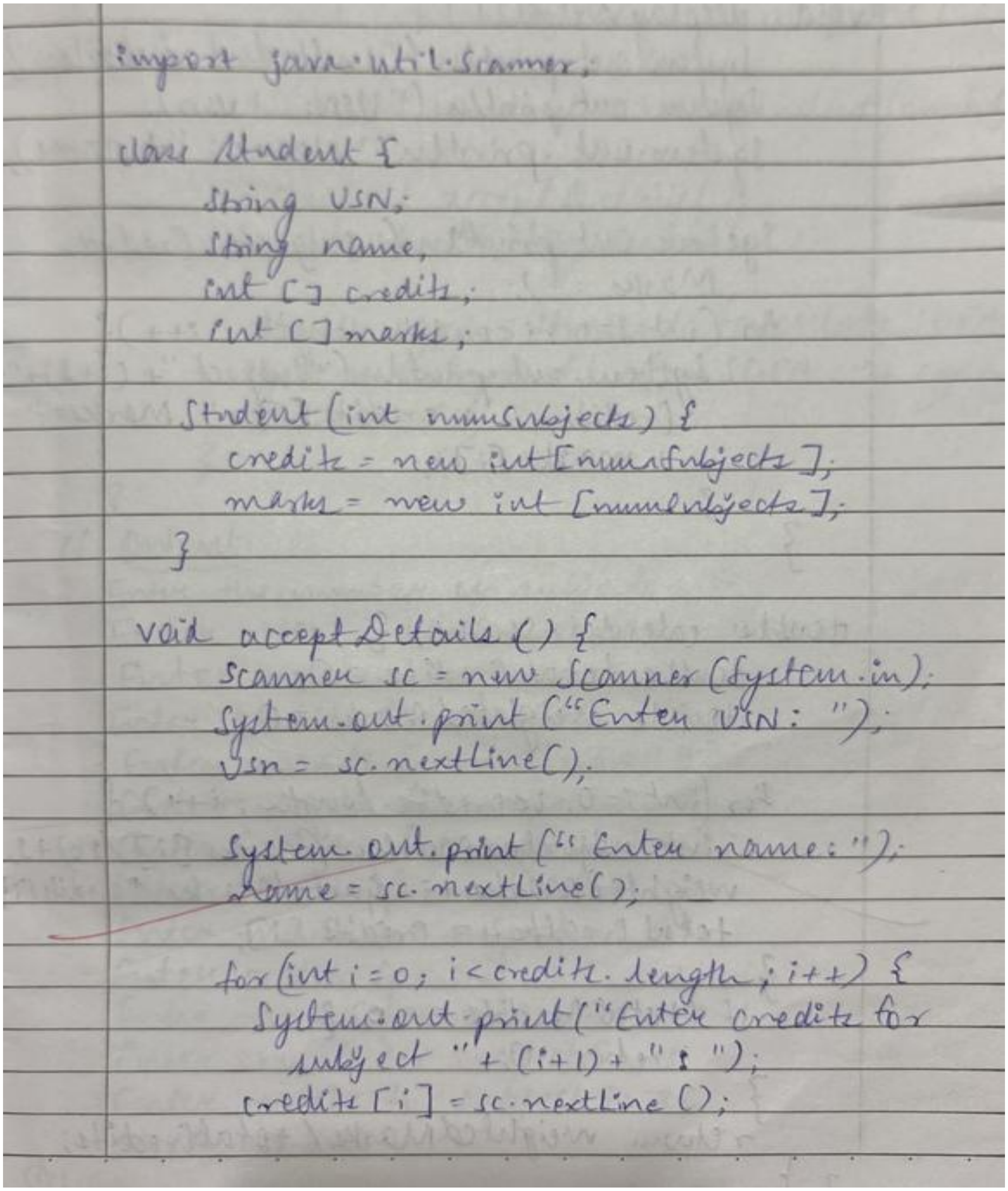


```
Enter coefficient a: 1
Enter coefficient b: 2
Enter coefficient c: 7
There are no real solutions; Discriminant is negative
UTKRISHT UMANG
1BM23ET056
```

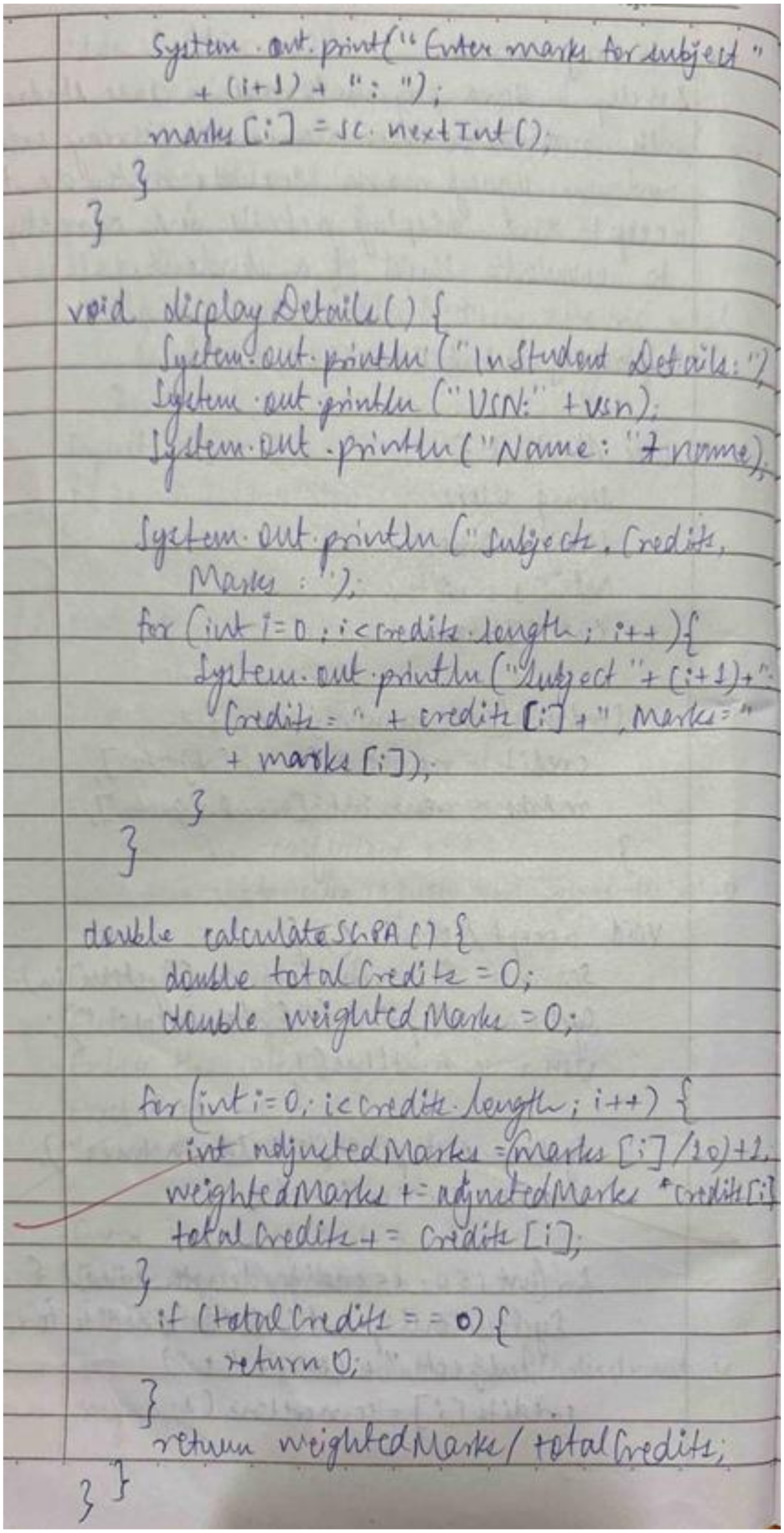
**Program 2**

Calculating SGPA

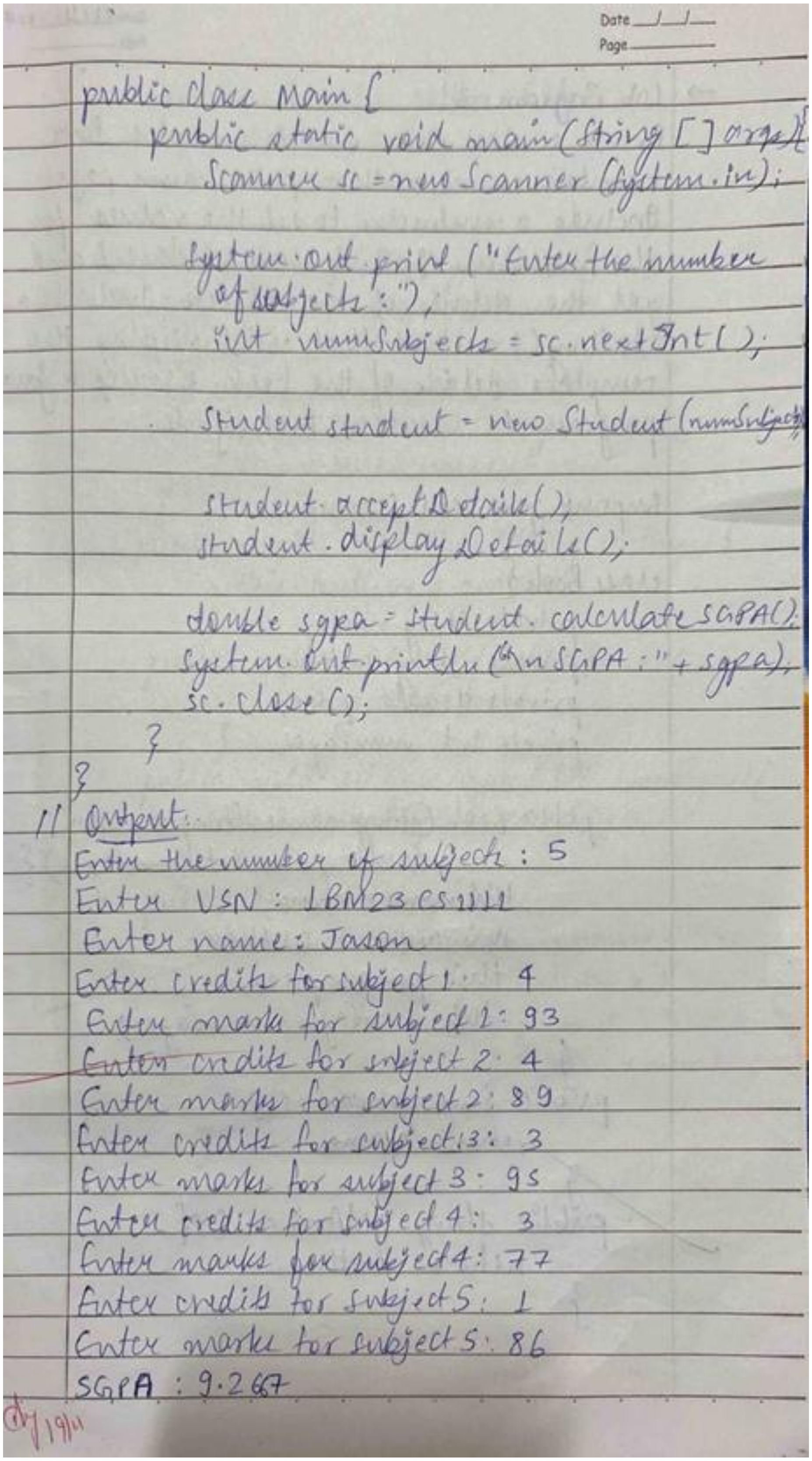
Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.



```
import java.util.Scanner;

class Student {
     String USN;
     String name,
     int [] credits;
     int [] marks;

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

 void acceptDetails() {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter USN: ");
      usn = sc.nextLine();

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

      for(int i=0; i<credits.length; i++) {
       System.out.print("Enter credits for
         subject "+ (i+1) + ": ");
       credits[i] = sc.nextLine();
```



```
        System.out.print("Enter marks for subject "
            + (i+1) + ": ");
        marks[i] = sc.nextInt();
      }
  }

void displayDetails() {
    System.out.println("\nStudent Details:");
    System.out.println("USN: " +usn);
    System.out.println("Name: " + name);

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

double calculateSGPA() {
     double totalCredits = 0;
     double weightedMarks = 0;

    for (int i=0; i<credits.length; i++) {
       int adjustedMarks = (marks[i]/10)+1;
       weightedMarks += adjustedMarks *credits[i]
       totalCredits += credits[i];
    }
     if (totalCredits == 0) {
         return 0;
     }
     return weightedMarks/totalCredits;
} }
```



Date _/_/_
Page____

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

        System.out.print ("Enter the number
         of subjects :");
        int numSubjects = sc.nextInt( );

        Student student = new Student (numSubjects

        student.acceptDetails();
        student.displayDetails();

        double sgpa = student.calculateSGPA();
        System.out.println("\n SGPA :" + sgpa);
        sc.close();
    }
}
```

// Output:

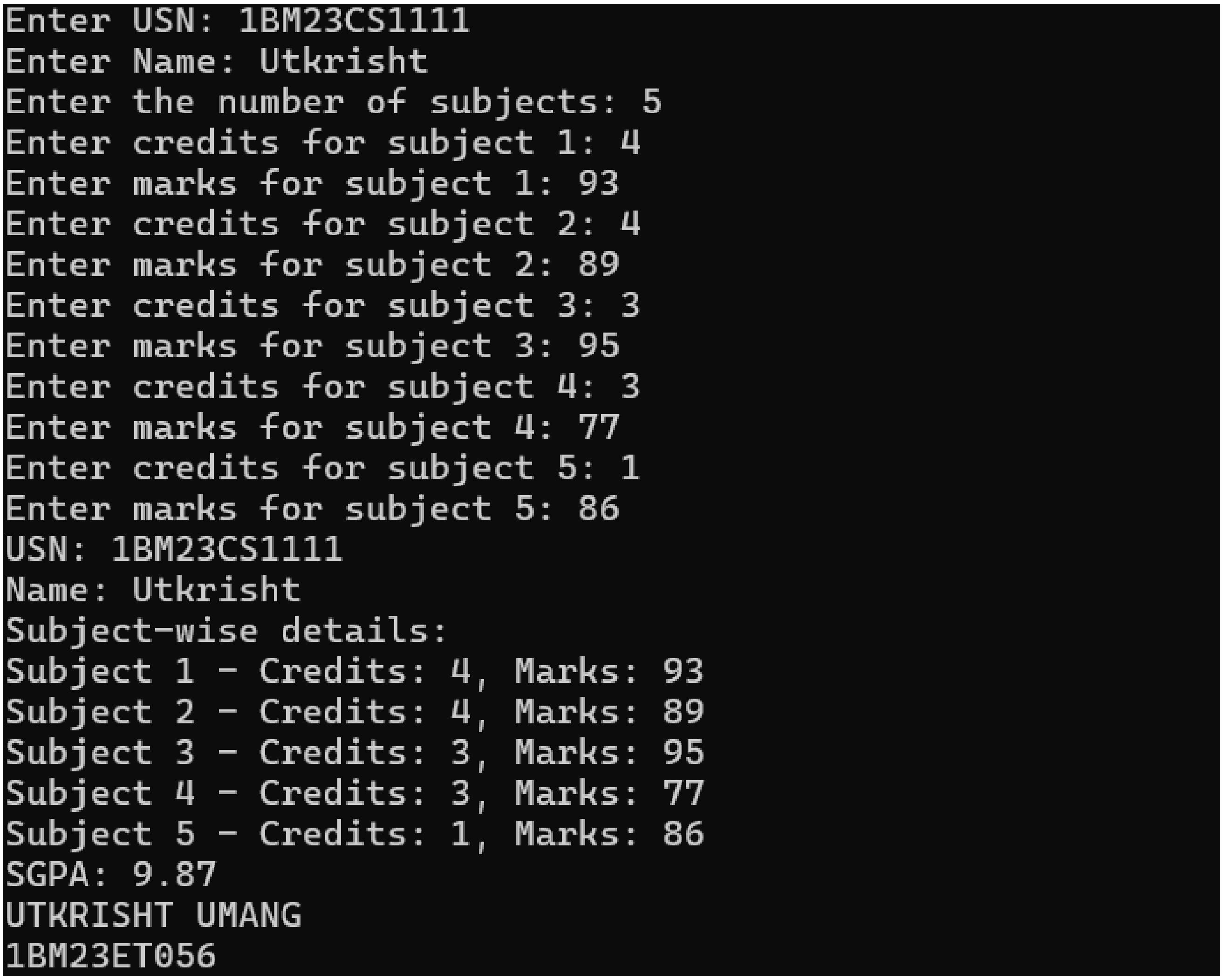
```
Enter the number of subjects : 5
Enter USN : 1BM23CS1111
Enter name: Jason
Enter credits for subject 1:  4
Enter marks for subject 1: 93
Enter credits for subject 2: 4
Enter marks for subject 2: 89
Enter credits for subject 3: 3
Enter marks for subject 3: 95
Enter credits for subject 4: 3
Enter marks for subject 4: 77
Enter credits for subject 5: 1
Enter marks for subject 5: 86
SGPA : 9.267
```

19/11

Code:

```
import java.util.Scanner;

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

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

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

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

        System.out.print("Enter the number of subjects: ");
        int numSubjects = sc.nextInt();

        credits = new int[numSubjects];
        marks = new int[numSubjects];

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

            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = sc.nextInt();
        }

        sc.close();
    }

    public void displayDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Subject-wise details:");
        for (int i = 0; i < credits.length; i++) {
            System.out.println("Subject " + (i + 1) + " - Credits: " + credits[i] + ", Marks: " + marks[i]);
        }
    }

    public double calculateSGPA() {
        double weightedMarks = 0;
        int totalCredits = 0;

        for (int i = 0; i < credits.length; i++) {
            double adjustedMarks = marks[i] / 10.0 + 1;
            weightedMarks += adjustedMarks * credits[i];
            totalCredits += credits[i];
        }

        return weightedMarks / totalCredits;
    }
}

public class Main {
    public static void main(String[] args) {
        Student student = new Student();

        student.acceptDetails();
        student.displayDetails();

        double sgpa = student.calculateSGPA();
        System.out.printf("SGPA: %.2f%n", sgpa);

        System.out.println("UTKRISHT UMANG\n1BM23ET056");
    }
}
```

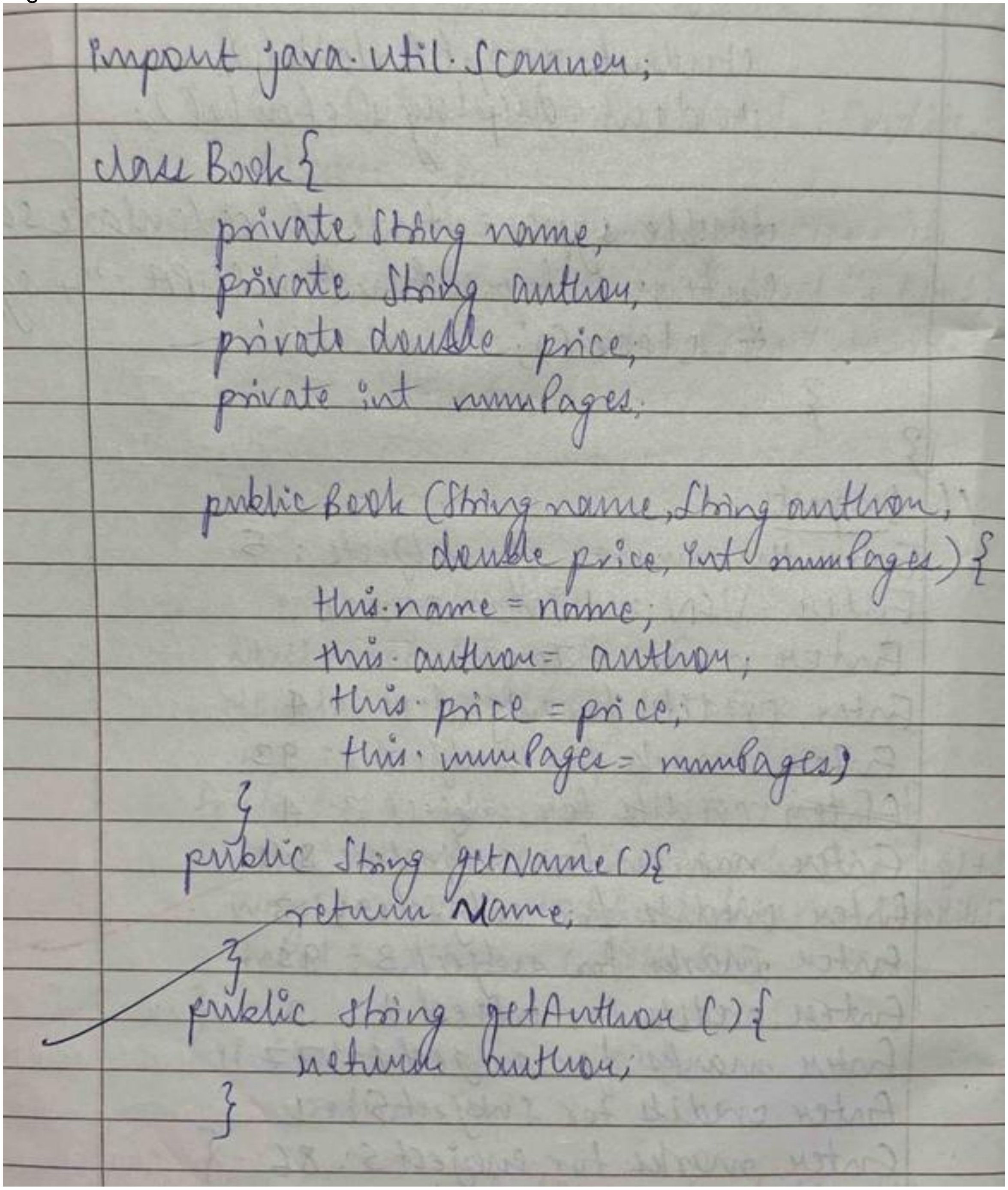
Output:



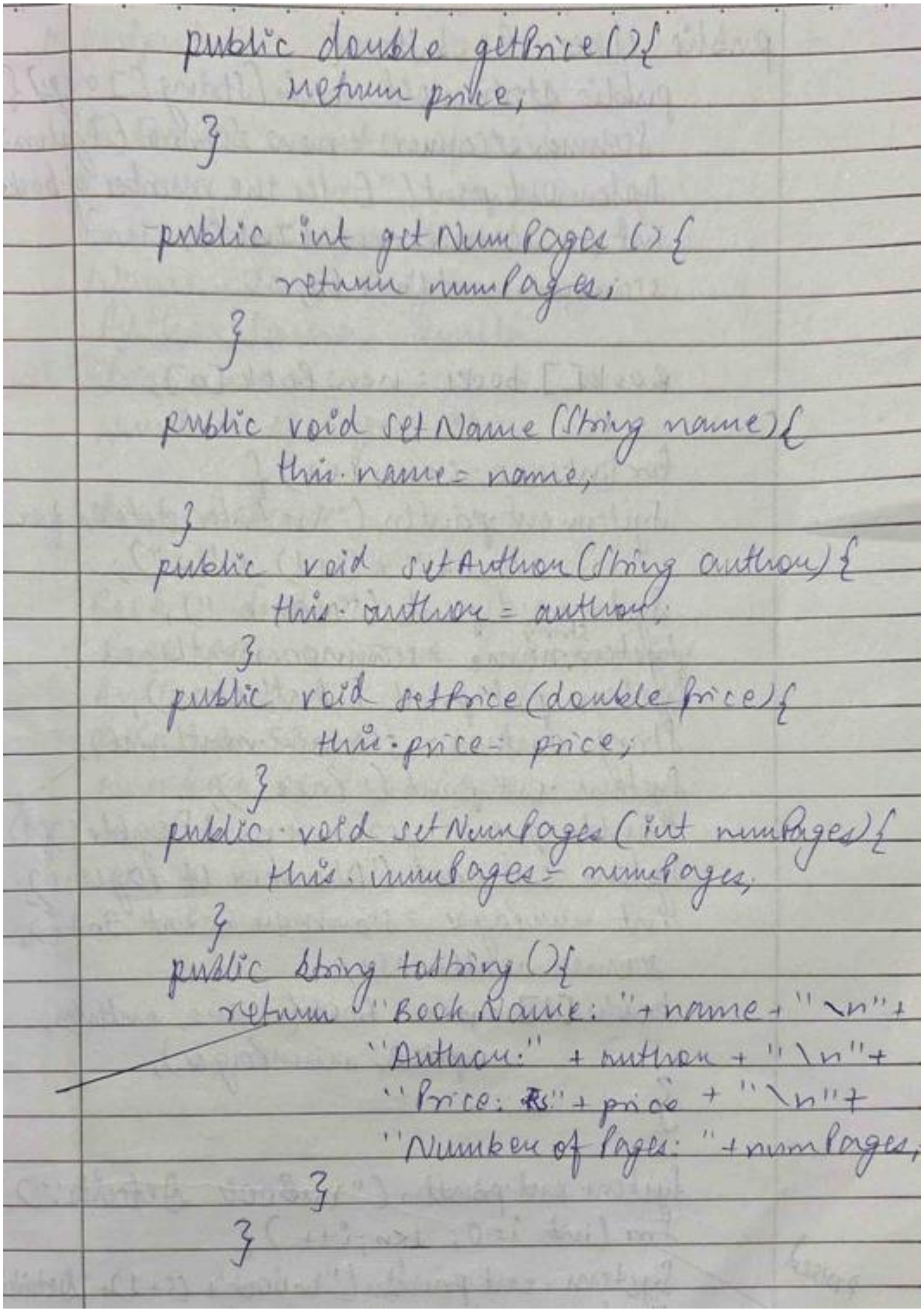
```
Enter USN: 1BM23CS1111
Enter Name: Utkrisht
Enter the number of subjects: 5
Enter credits for subject 1: 4
Enter marks for subject 1: 93
Enter credits for subject 2: 4
Enter marks for subject 2: 89
Enter credits for subject 3: 3
Enter marks for subject 3: 95
Enter credits for subject 4: 3
Enter marks for subject 4: 77
Enter credits for subject 5: 1
Enter marks for subject 5: 86
USN: 1BM23CS1111
Name: Utkrisht
Subject-wise details:
Subject 1 - Credits: 4, Marks: 93
Subject 2 - Credits: 4, Marks: 89
Subject 3 - Credits: 3, Marks: 95
Subject 4 - Credits: 3, Marks: 77
Subject 5 - Credits: 1, Marks: 86
SGPA: 9.87
UTKRISHT UMANG
1BM23ET056
```

**Program 3**

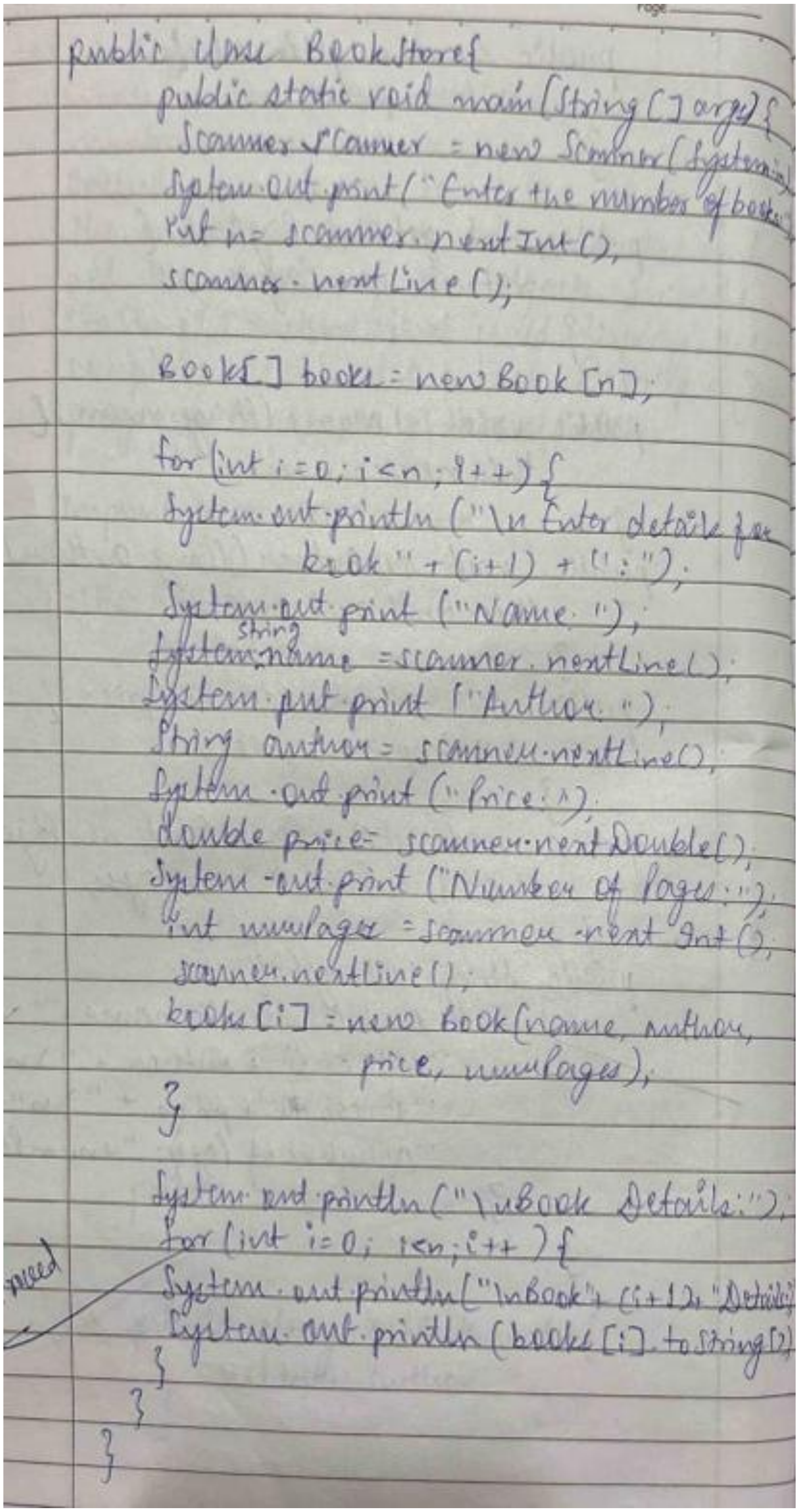
Book Details
Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a tostring( ) method that could display the complete details of the book. Develop a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



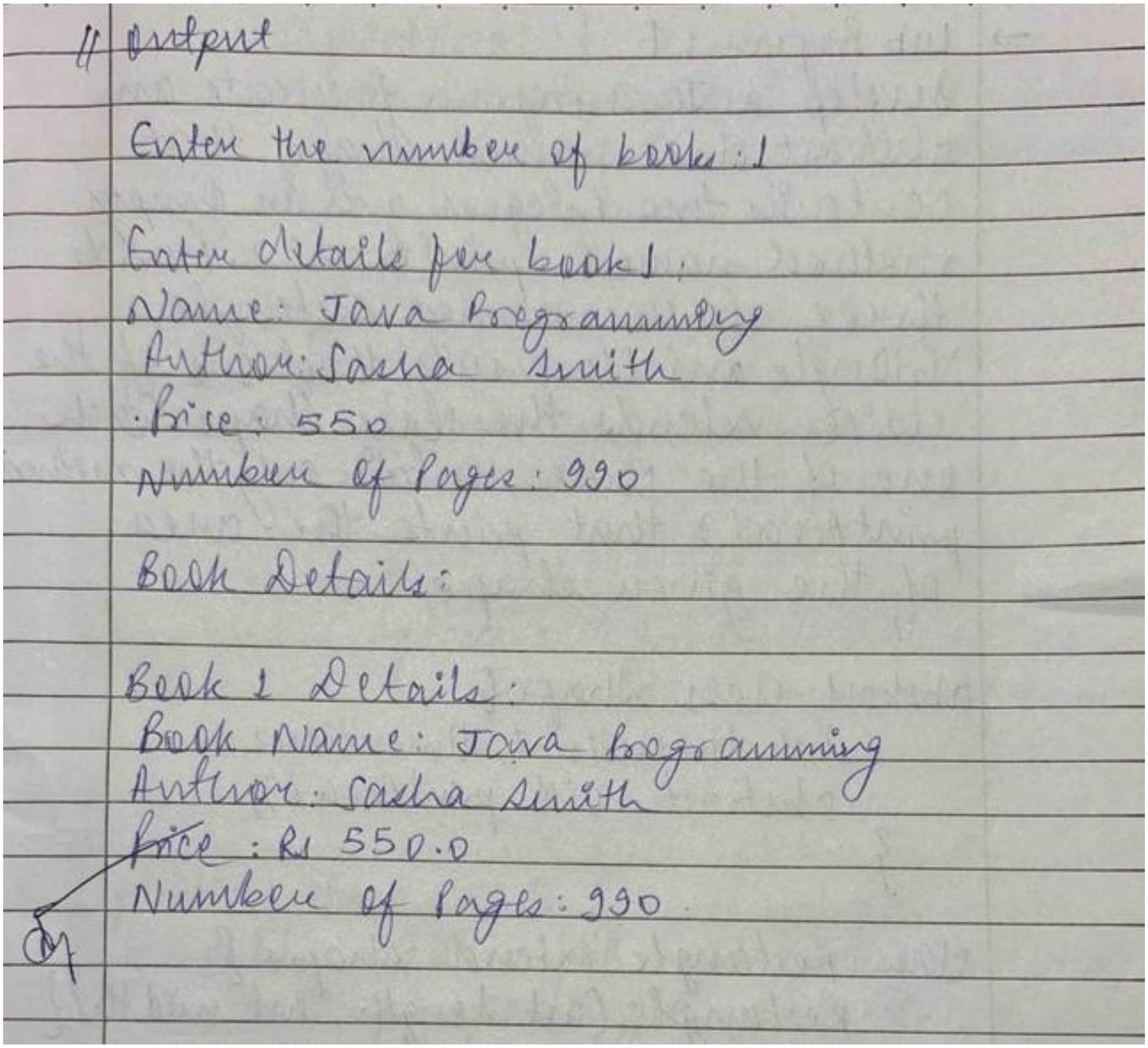
```
Import java.util.Scanner;

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

        public Book(String name, String author,
                        double price, int numPages){
            this.name = name;
            this.author= author;
            this.price = price;
            this.numPages= numPages;
        }
      public String getName(){
            return Name;
        }
      public String getAuthor(){
            return author;
        }
```



```
public double getPrice(){
        return price;
  }

public int getNumPages(){
        return numPages;
   }

public void setName(String name){
        this.name= name;
 }
public void setAuthor(String author){
        this.author = author;
    }
public void setPrice(double price){
        this.price= price;
     }
public void setNumPages(int numPages){
        this.numPages= numPages;
  }
public String toString(){
    return "Book Name: "+name+"\n"+
           "Author:" + author + "\n"+
           "Price: Rs"+ price + "\n"+
           "Number of Pages: "+numPages;
       }
  }
```



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

    Book[] books = new Book[n];

    for (int i=0; i<n; i++){
    System.out.println("\n Enter details for
              book" + (i+1) + ":");
    System.out.print("Name: ");
   ~~System~~String name =scanner.nextLine();
   System.out.print ("Author: ");
   String author= scanner.nextLine();
   System.out.print ("Price: ");
   double price= scanner.nextDouble();
   System.out.print ("Number of Pages: ");
    int numPages =scanner.nextInt();
     scanner.nextLine();
    books[i] =new Book(name, author,
                 price, numPages);
    }

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

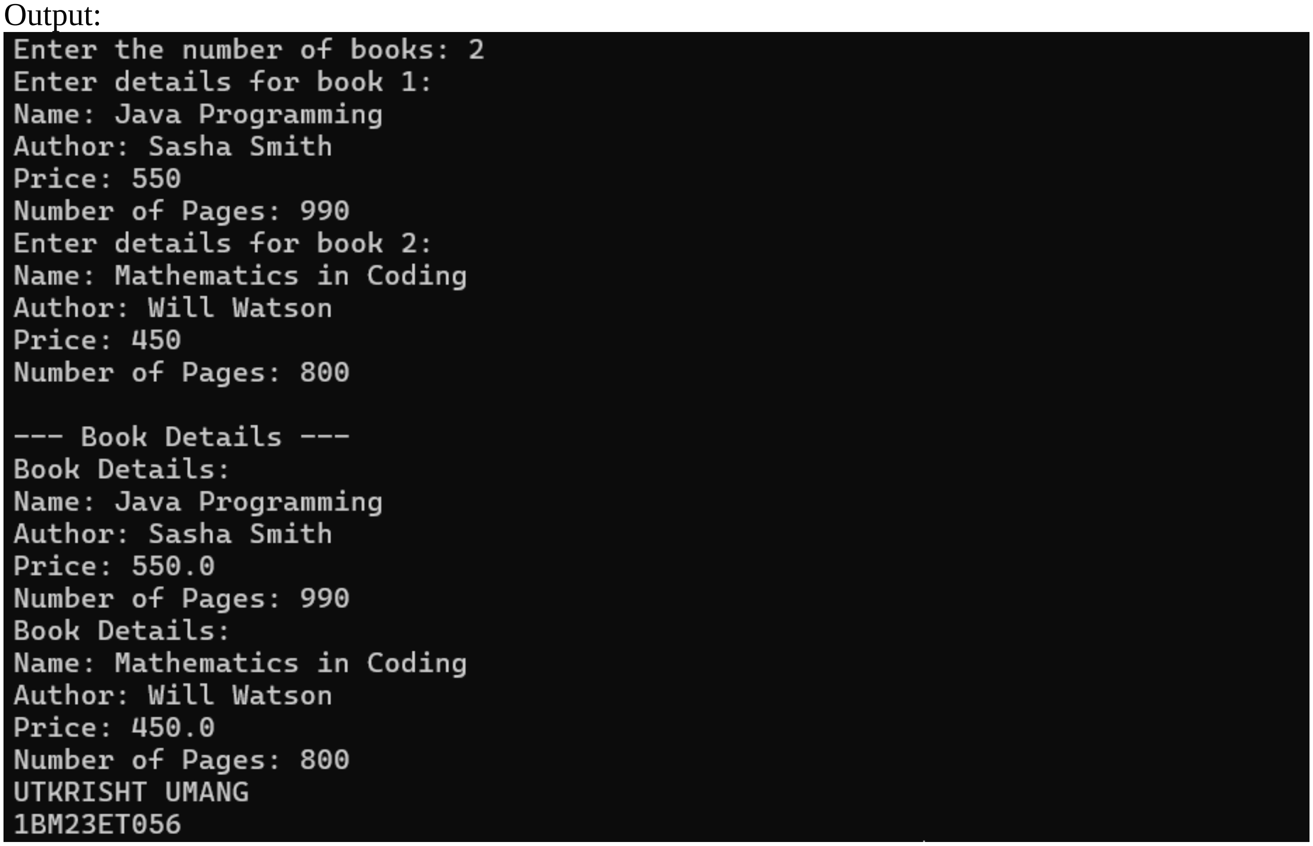


```
// Output
Enter the number of books: 1

Enter details for book 1:
Name: Java Programming
Author: Sasha Smith
Price: 550
Number of Pages: 990

Book Details:

Book 1 Details:
Book Name: Java Programming
Author: Sasha Smith
Price: Rs 550.0
Number of Pages: 990
```

Code:

```
import java.util.Scanner;

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

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

    public void setDetails(String name, String author, double price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

    public String getDetails() {
        return "Name: " + name + ", Author: " + author + ", Price: " + price + ", Pages: " + num_pages;
    }

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

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

        System.out.print("Enter the number of books: ");
        int n = sc.nextInt();
        sc.nextLine(); // Consume the leftover newline
        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for book " + (i + 1) + ":");
            System.out.print("Name: ");
            String name = sc.nextLine();
            System.out.print("Author: ");
            String author = sc.nextLine();
            System.out.print("Price: ");
            double price = sc.nextDouble();
            System.out.print("Number of Pages: ");
            int num_pages = sc.nextInt();
            sc.nextLine(); // Consume the leftover newline
            books[i] = new Book(name, author, price, num_pages);
        }
        System.out.println("\n--- Book Details ---");
        for (int i = 0; i < n; i++) {
            System.out.println(books[i].toString());
        }

        System.out.println("UTKRISHT UMANG\n1BM23ET056");
        sc.close();
    }
}
```

Output:



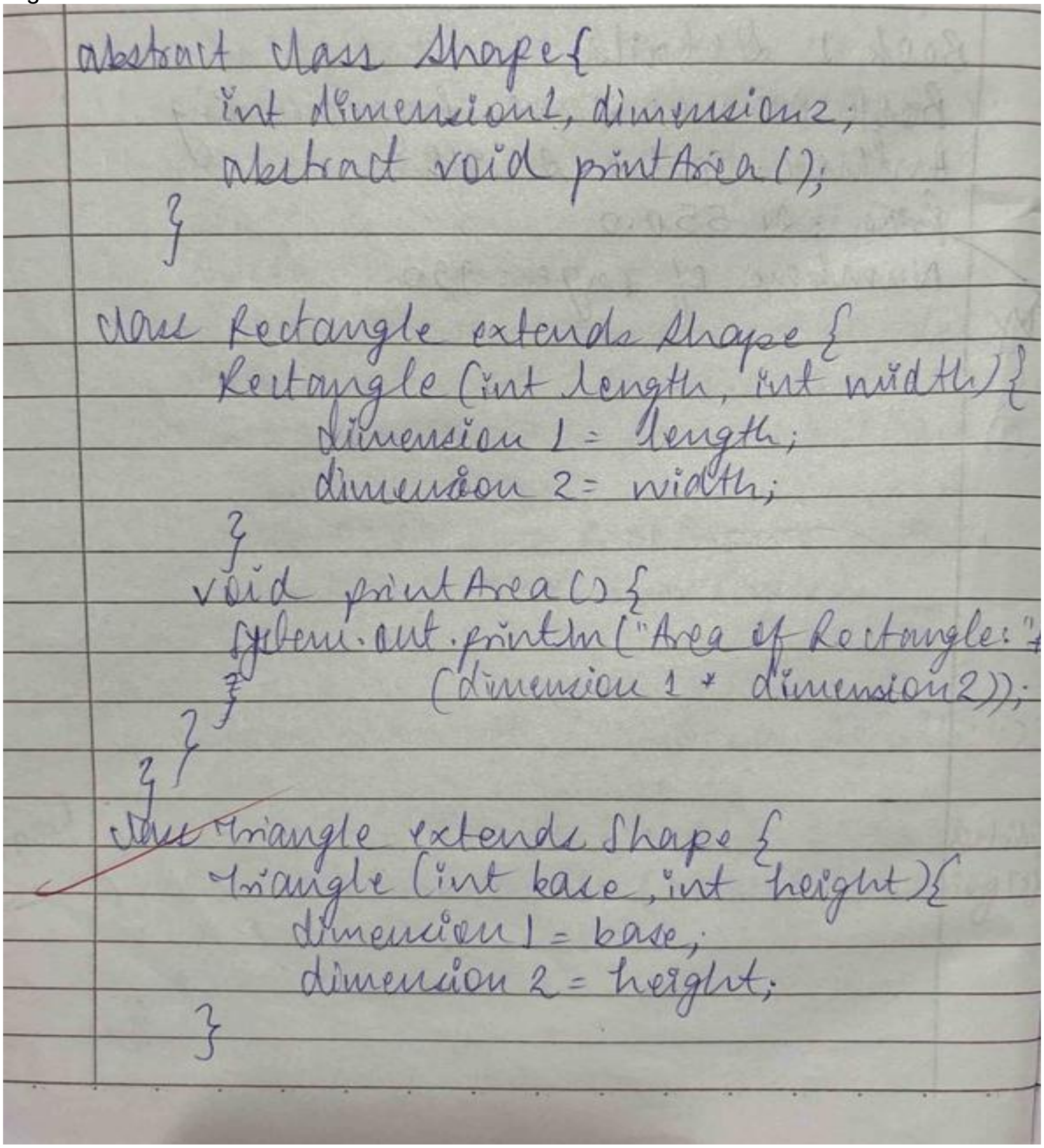
```
Enter the number of books: 2
Enter details for book 1:
Name: Java Programming
Author: Sasha Smith
Price: 550
Number of Pages: 990
Enter details for book 2:
Name: Mathematics in Coding
Author: Will Watson
Price: 450
Number of Pages: 800

--- Book Details ---
Book Details:
Name: Java Programming
Author: Sasha Smith
Price: 550.0
Number of Pages: 990
Book Details:
Name: Mathematics in Coding
Author: Will Watson
Price: 450.0
Number of Pages: 800
UTKRISHT UMANG
1BM23ET056
```

**Program 4**

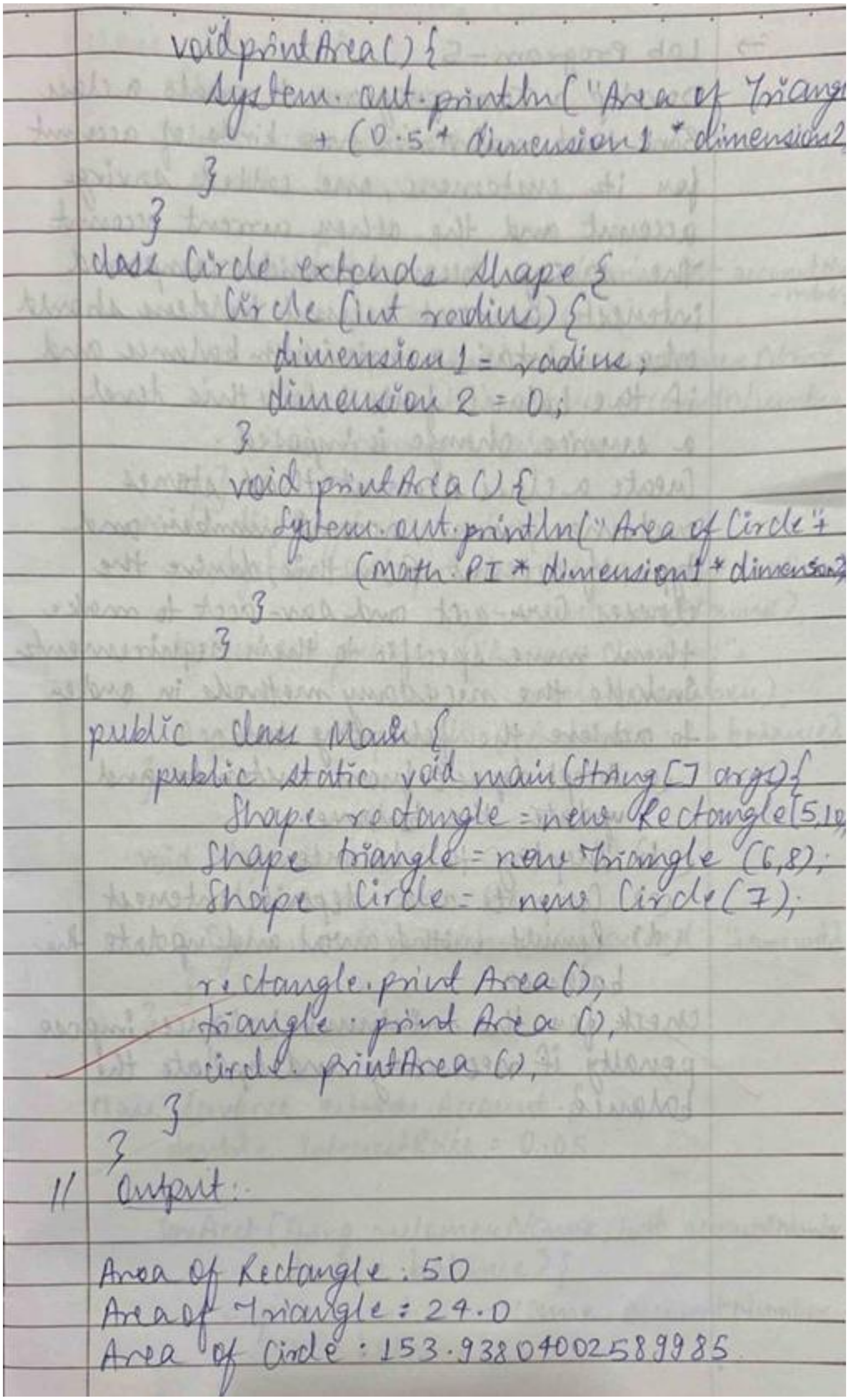
Abstract Class Shape

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



```
abstract class Shape{
      int dimension1, dimension2;
      abstract void printArea();
   }

class Rectangle extends Shape {
      Rectangle (int length, int width){
            dimension 1 = length;
            dimension 2= width;
      }
    void printArea(){
     System.out.println("Area of Rectangle:"+
                (dimension 1 * dimension2));
     }
   }
 }
class Triangle extends Shape {
     Triangle (int base, int height){
          dimension 1 = base;
          dimension 2 = height;
     }
```



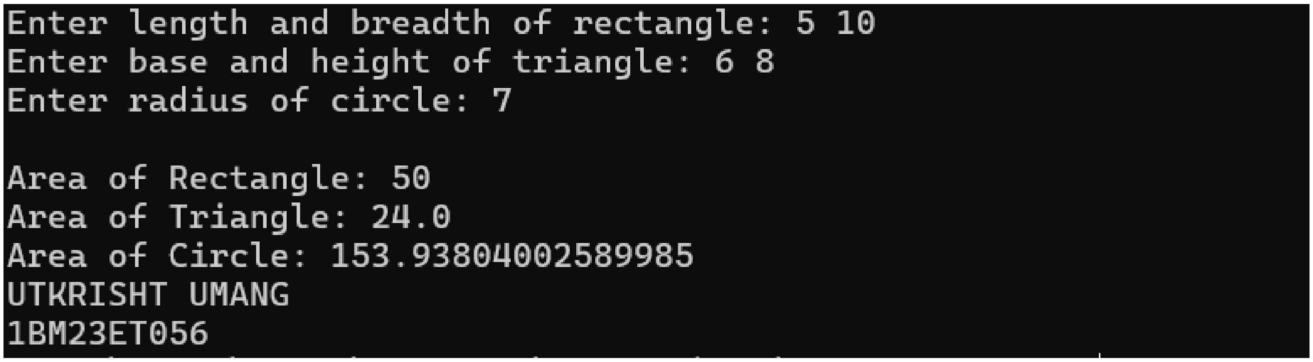
```
        voidprintArea(){
            System.out.println("Area of Triang
                + (0.5 * dimension1 * dimension2
        }
    }
class Circle extends Shape {
        Circle (int radius){
            dimension1 = radius;
            dimension2 = 0;
        }
        void printArea(){
            System.out.println("Area of Circle"+
                (Math.PI * dimension1 * dimension2
        }
    }

public class Main {
    public static void main(String[] args){
        Shape rectangle = new Rectangle[5,10
        Shape triangle = new Triangle (6,8);
        Shape Circle = new Circle(7);

        rectangle.print Area();
        triangle.print Area();
        circle.printArea();
    }
}
// Output:
```

```
Area of Rectangle: 50
Area of Triangle: 24.0
Area of Circle: 153.93804002589985
```

Code:

```
import java.util.Scanner;

abstract class Shape {
    int dimension1;
    int dimension2;

    abstract void printArea();
}

class Rectangle extends Shape {
    Rectangle(int length, int breadth) {
        this.dimension1 = length;
        this.dimension2 = breadth;
    }

    @Override
    void printArea() {
        System.out.println("Area of Rectangle: " + (dimension1 * dimension2));
    }
}

class Triangle extends Shape {
    Triangle(int base, int height) {
        this.dimension1 = base;
        this.dimension2 = height;
    }

    @Override
    void printArea() {
        System.out.println("Area of Triangle: " + (0.5 * dimension1 * dimension2));
    }
}

class Circle extends Shape {
    Circle(int radius) {
        this.dimension1 = radius;
    }

    @Override
    void printArea() {
        System.out.println("Area of Circle: " + (Math.PI * dimension1 * dimension1));
    }
}

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

        System.out.print("Enter length and breadth of rectangle: ");
        int length = sc.nextInt();
        int breadth = sc.nextInt();
        Shape rectangle = new Rectangle(length, breadth);

        System.out.print("Enter base and height of triangle: ");
        int base = sc.nextInt();
        int height = sc.nextInt();
        Shape triangle = new Triangle(base, height);

        System.out.print("Enter radius of circle: ");
        int radius = sc.nextInt();
        Shape circle = new Circle(radius);

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

        System.out.println("UTKRISHT UMANG\n1BM23ET056");
        sc.close();
    }
}
```

Output:



```
Enter length and breadth of rectangle: 5 10
Enter base and height of triangle: 6 8
Enter radius of circle: 7

Area of Rectangle: 50
Area of Triangle: 24.0
Area of Circle: 153.93804002589985
UTKRISHT UMANG
1BM23ET056
```

**Program 5**

Bank Details

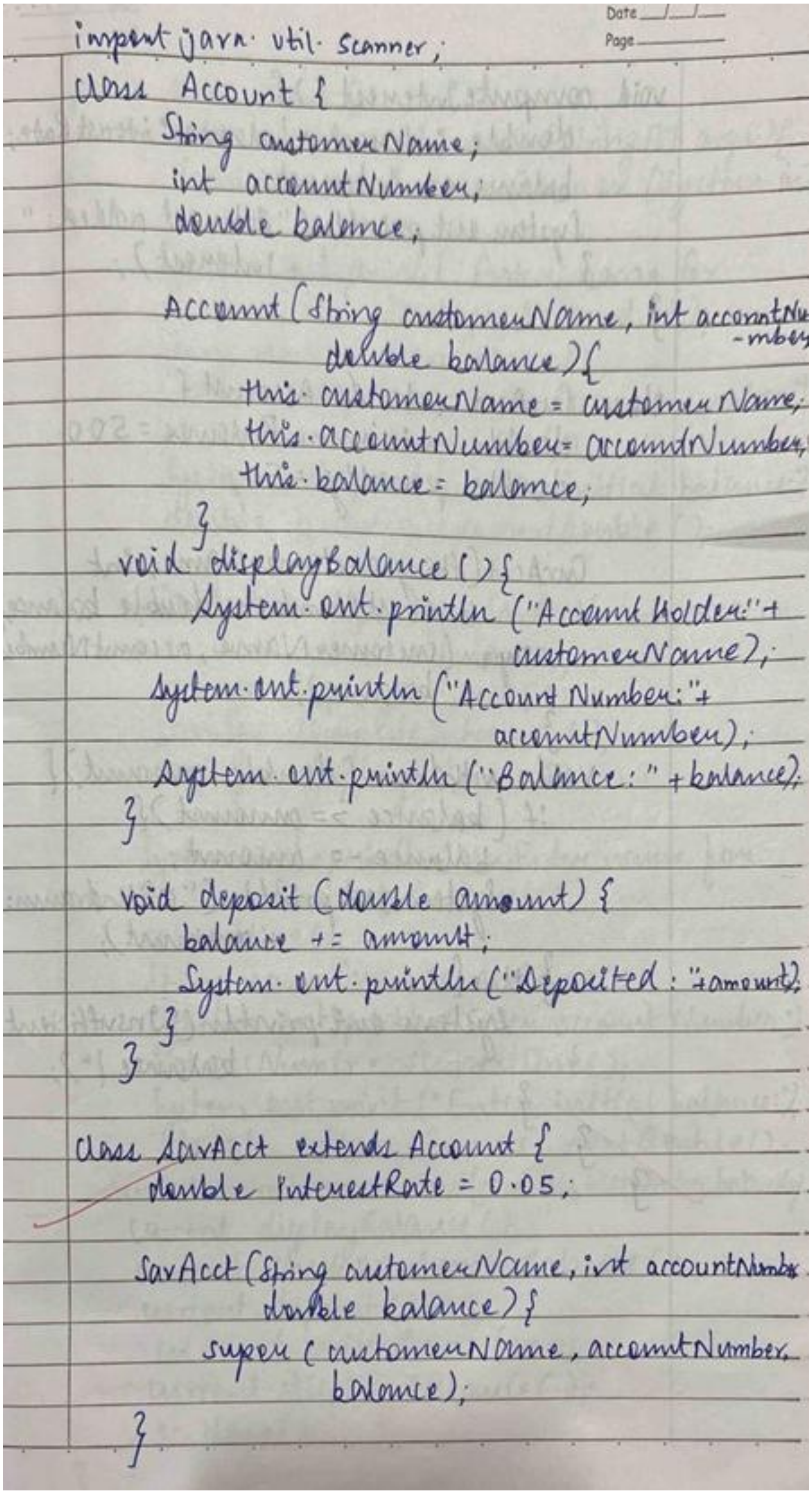
Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book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

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

a) Accept deposit from customer and update the balance.
b) Display the balance.
c) Compute and deposit interest

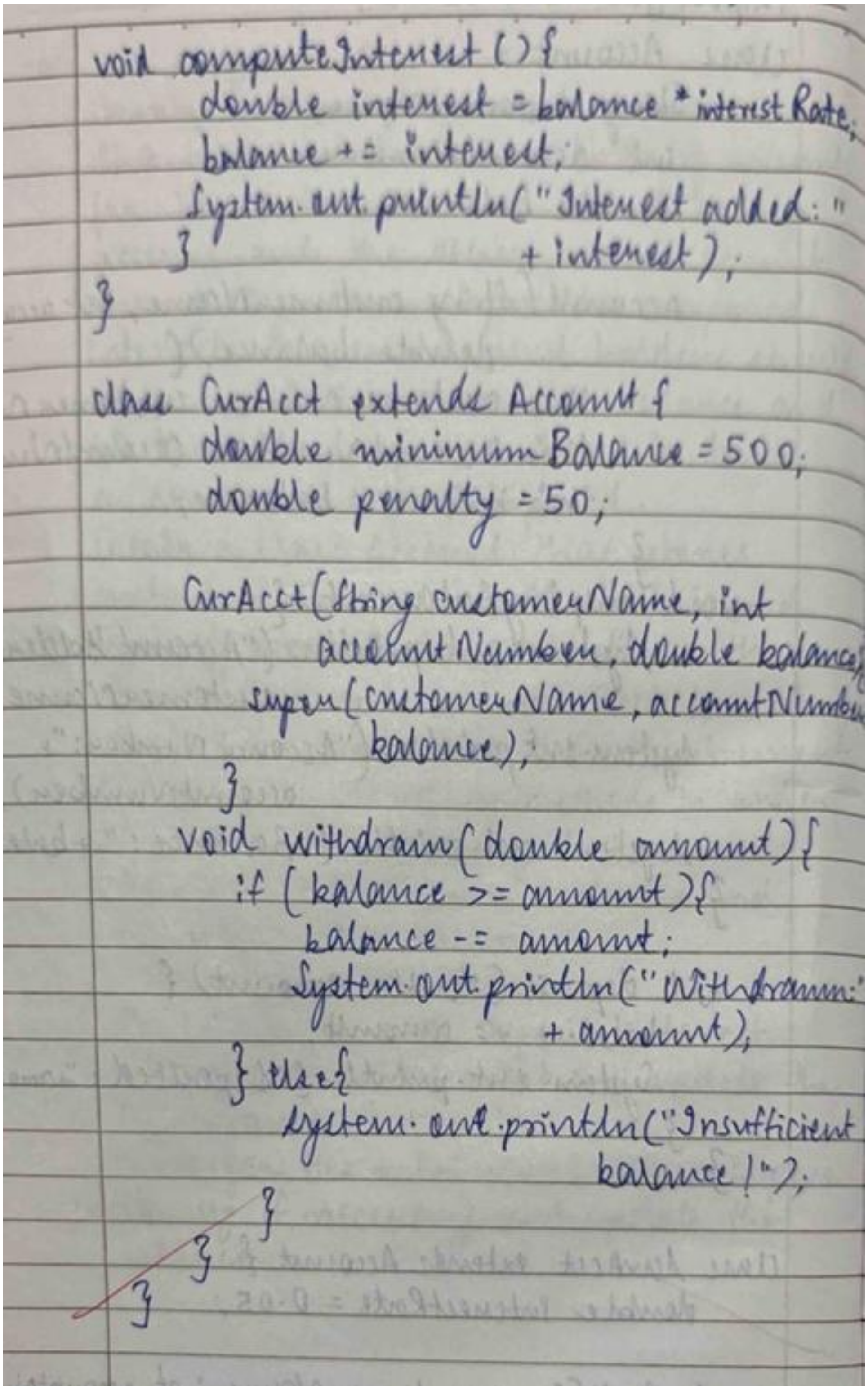
Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

Algorithm:

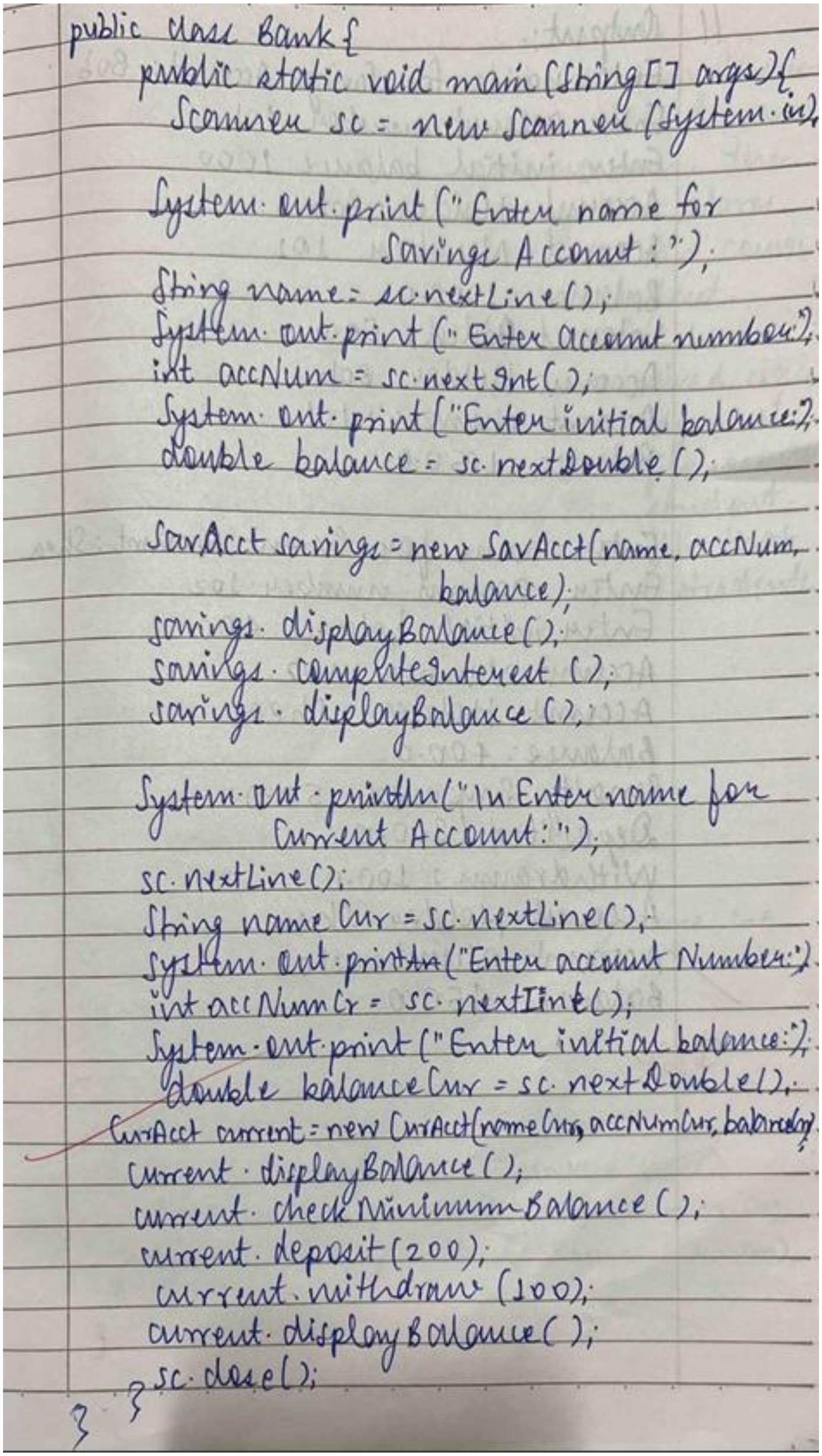


Date __/__/__
Page ____

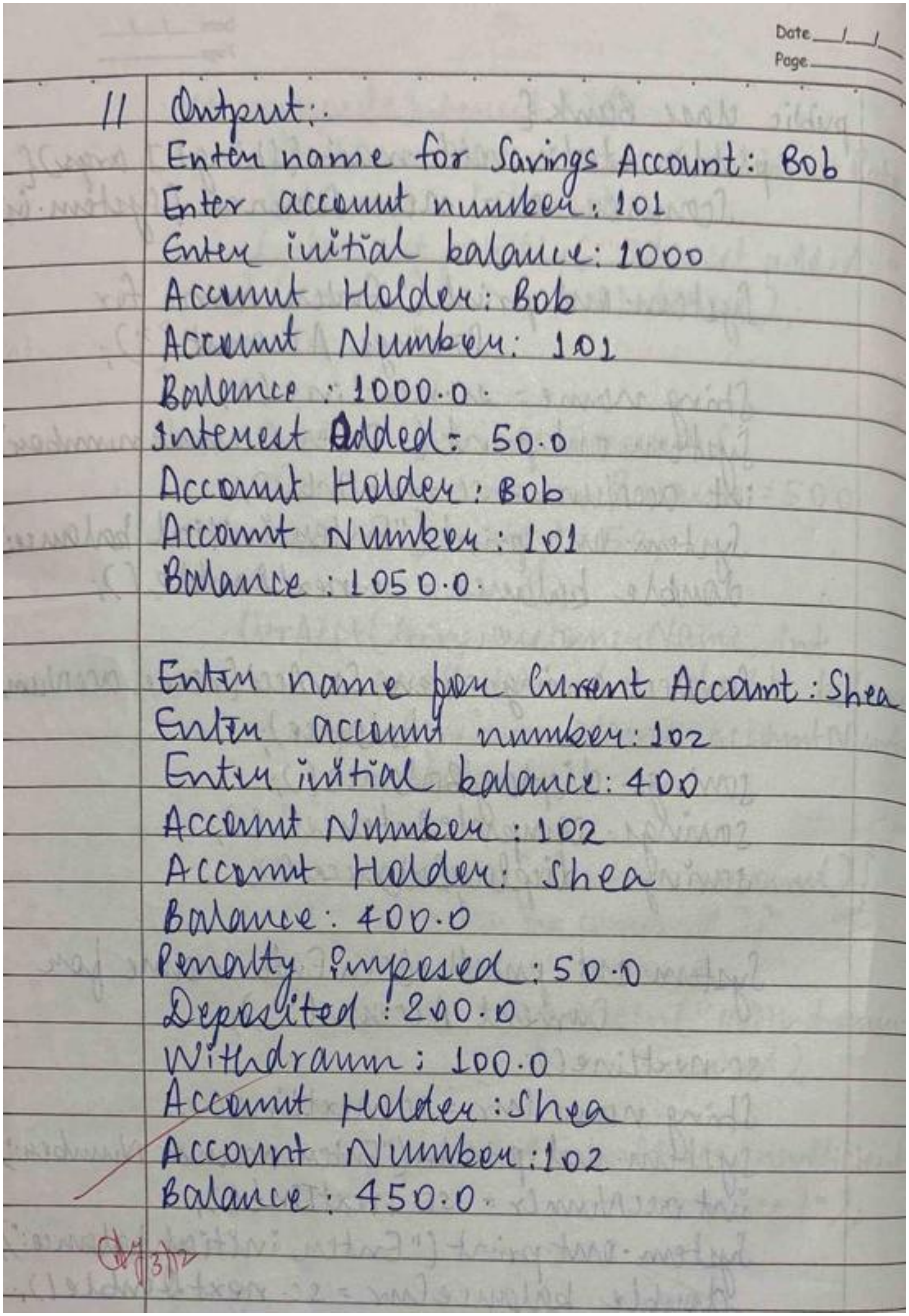
```
import java. util. Scanner;
class Account {
    String customerName;
    int accountNumber;
    double balance;

    Account(String customerName, int accountNumber,
                double balance){
        this.customerName = customerName;
        this.accountNumber= accountNumber;
        this.balance = balance;
    }
 void displayBalance(){
     System.out.println("Account Holder:"+
                          customerName);
   System.out.println("Account Number:"+
                          accountNumber);
    System.out.println("Balance:" +balance);
 }

 void deposit(double amount) {
    balance += amount;
     System.out.println("Deposited :"+amount);
 }
 }

class SavAcct extends Account {
  double InterestRate = 0.05;

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



```
void computeInterest (){
        double interest = balance * interest Rate;
        balance += interest;
        System.out.println("Interest added:"
      }                        + interest);
}

class CurAcct extends Account {
        double minimumBalance = 500;
        double penalty = 50;

        CurAcct(String customerName, int
                accountNumber, double balance){
            super(customerName, accountNumber
                    balance);
        }
        void withdraw(double amount){
            if (balance >= amount){
                balance -= amount;
                System.out.println("Withdrawn:"
                                + amount);
            } else{
                System.out.println("Insufficient
                                balance!");
            }
        }
    }
```



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

      System.out.print("Enter name for
                         Savings Account:");
      String name= sc.nextLine();
      System.out.print(" Enter account number");
      int accNum = sc.nextInt();
      System.out.print("Enter initial balance:");
      double balance = sc.nextDouble();

      SavAcct savings = new SavAcct(name, accNum,
                            balance);
      savings.displayBalance();
      savings.computeInterest ();
      savings.displayBalance ();

      System.out.println("\n Enter name for
                 Current Account:");
      sc.nextLine();
      String nameCur = sc.nextLine();
      System.out.println("Enter account Number:");
      int accNumCr = sc.nextIint();
      System.out.print("Enter initial balance:");
       double balanceCur = sc.nextDouble();
     CurAcct current = new CurAcct(nameCur, accNumCur, balanceCur);
     current.displayBalance();
     current.checkMinimumBalance();
     current.deposit(200);
     current.withdraw(100);
     current.displayBalance();
    sc.close();
  }
}
```



Date / /
Page

```
// Output:
Enter name for Savings Account: Bob
Enter account number: 101
Enter initial balance: 1000
Account Holder: Bob
Account Number: 101
Balance : 1000.0.
Interest Added: 50.0
Account Holder: Bob
Account Number: 101
Balance : 1050.0.

Enter name for Current Account: Shea
Enter account number: 102
Enter initial balance: 400
Account Number : 102
Account Holder: Shea
Balance: 400.0
Penalty Imposed: 50.0
Deposited: 200.0
Withdrawn: 100.0
Account Holder: Shea
Account Number: 102
Balance: 450.0.
```

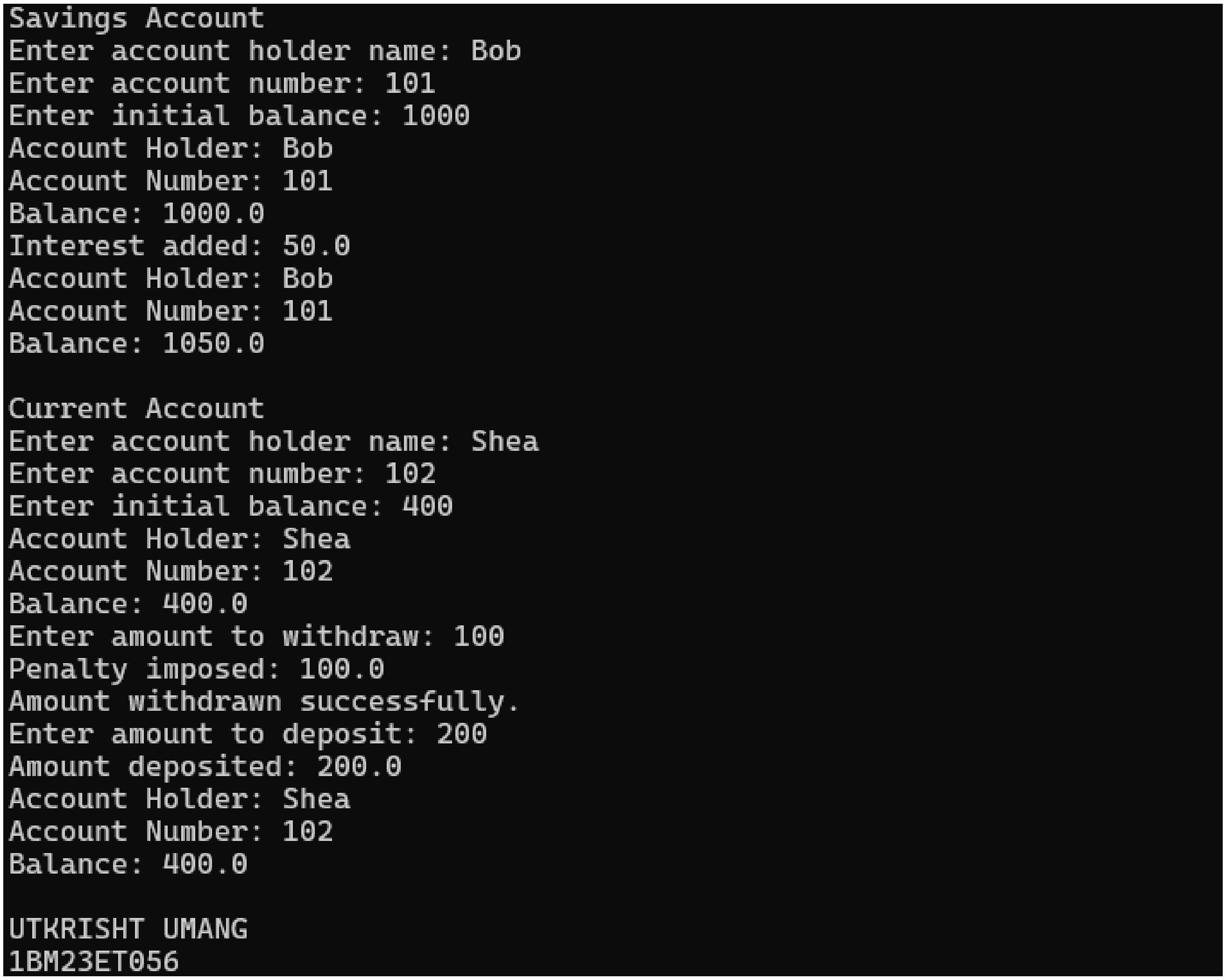
3/12

Code:

```
import java.util.Scanner;

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

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

    void deposit(double amount) {
        balance += amount;
    }

    void displayBalance() {
        System.out.println("Balance: " + balance);
    }
}

class SavAcct extends Account {
    private static final double INTEREST_RATE = 0.05;

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

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

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Amount withdrawn successfully.");
        } else {
            System.out.println("Insufficient balance for withdrawal.");
        }
    }
```

```
}

class CurAcct extends Account {
    private static final double MINIMUM_BALANCE = 1000;
    private static final double PENALTY = 100;

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

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            if (balance < MINIMUM_BALANCE) {
                balance -= PENALTY;
                System.out.println("Penalty imposed: " + PENALTY);
            }
            System.out.println("Amount withdrawn successfully.");
        } else {
            System.out.println("Insufficient balance for withdrawal.");
        }
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Amount deposited: " + amount);
    }
}

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

        System.out.println("Savings Account");
        System.out.print("Enter account holder name: ");
        String savName = sc.nextLine();
        System.out.print("Enter account number: ");
        String savAccNo = sc.nextLine();
        System.out.print("Enter initial balance: ");
        double savBalance = sc.nextDouble();
        SavAcct savingsAccount = new SavAcct(savName, savAccNo, savBalance);

        System.out.println("Account Holder: " + savingsAccount.customerName);
        System.out.println("Account Number: " + savingsAccount.accountNumber);
        System.out.println("Balance: " + savingsAccount.balance);
        savingsAccount.computeAndDepositInterest();
        System.out.println("Account Holder: " + savingsAccount.customerName);
        System.out.println("Account Number: " + savingsAccount.accountNumber);
        System.out.println("Balance: " + savingsAccount.balance);

        System.out.println("\nCurrent Account");
        sc.nextLine();
        System.out.print("Enter account holder name: ");
        String curName = sc.nextLine();
        System.out.print("Enter account number: ");
        String curAccNo = sc.nextLine();
        System.out.print("Enter initial balance: ");
        double curBalance = sc.nextDouble();
        CurAcct currentAccount = new CurAcct(curName, curAccNo, curBalance);

        System.out.println("Account Holder: " + currentAccount.customerName);
        System.out.println("Account Number: " + currentAccount.accountNumber);
        System.out.println("Balance: " + currentAccount.balance);
        System.out.print("Enter amount to withdraw: ");
        double withdrawAmount = sc.nextDouble();
        currentAccount.withdraw(withdrawAmount);

        System.out.print("Enter amount to deposit: ");
        double depositAmount = sc.nextDouble();
        currentAccount.deposit(depositAmount);

        System.out.println("Account Holder: " + currentAccount.customerName);
        System.out.println("Account Number: " + currentAccount.accountNumber);
        System.out.println("Balance: " + currentAccount.balance);

        System.out.println("\nUTKRISHT UMANG\n1BM23ET056");
        sc.close();
    }
}
```

Output:



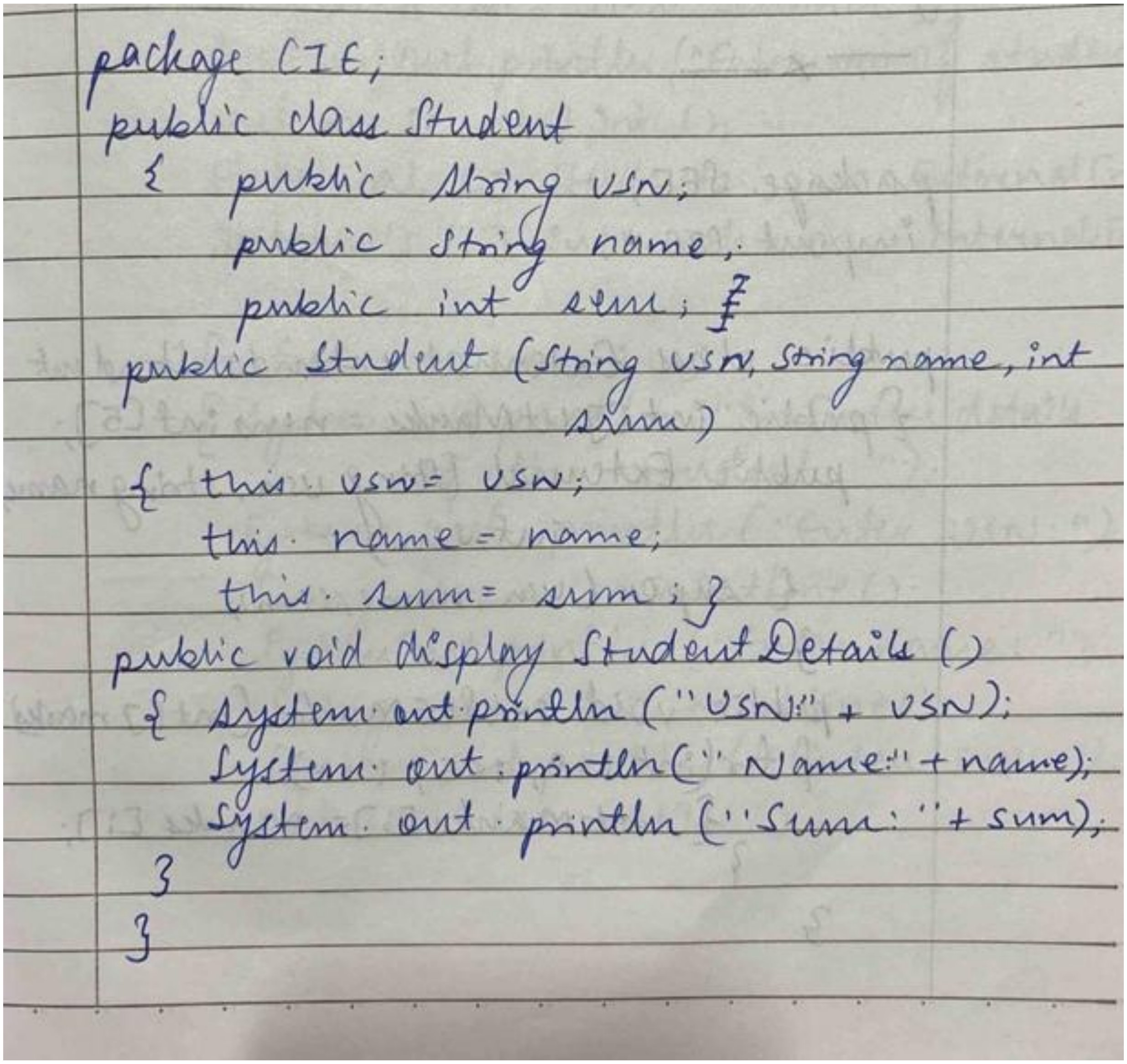
```
Savings Account
Enter account holder name: Bob
Enter account number: 101
Enter initial balance: 1000
Account Holder: Bob
Account Number: 101
Balance: 1000.0
Interest added: 50.0
Account Holder: Bob
Account Number: 101
Balance: 1050.0

Current Account
Enter account holder name: Shea
Enter account number: 102
Enter initial balance: 400
Account Holder: Shea
Account Number: 102
Balance: 400.0
Enter amount to withdraw: 100
Penalty imposed: 100.0
Amount withdrawn successfully.
Enter amount to deposit: 200
Amount deposited: 200.0
Account Holder: Shea
Account Number: 102
Balance: 400.0

UTKRISHT UMANG
1BM23ET056
```

**Program 6**

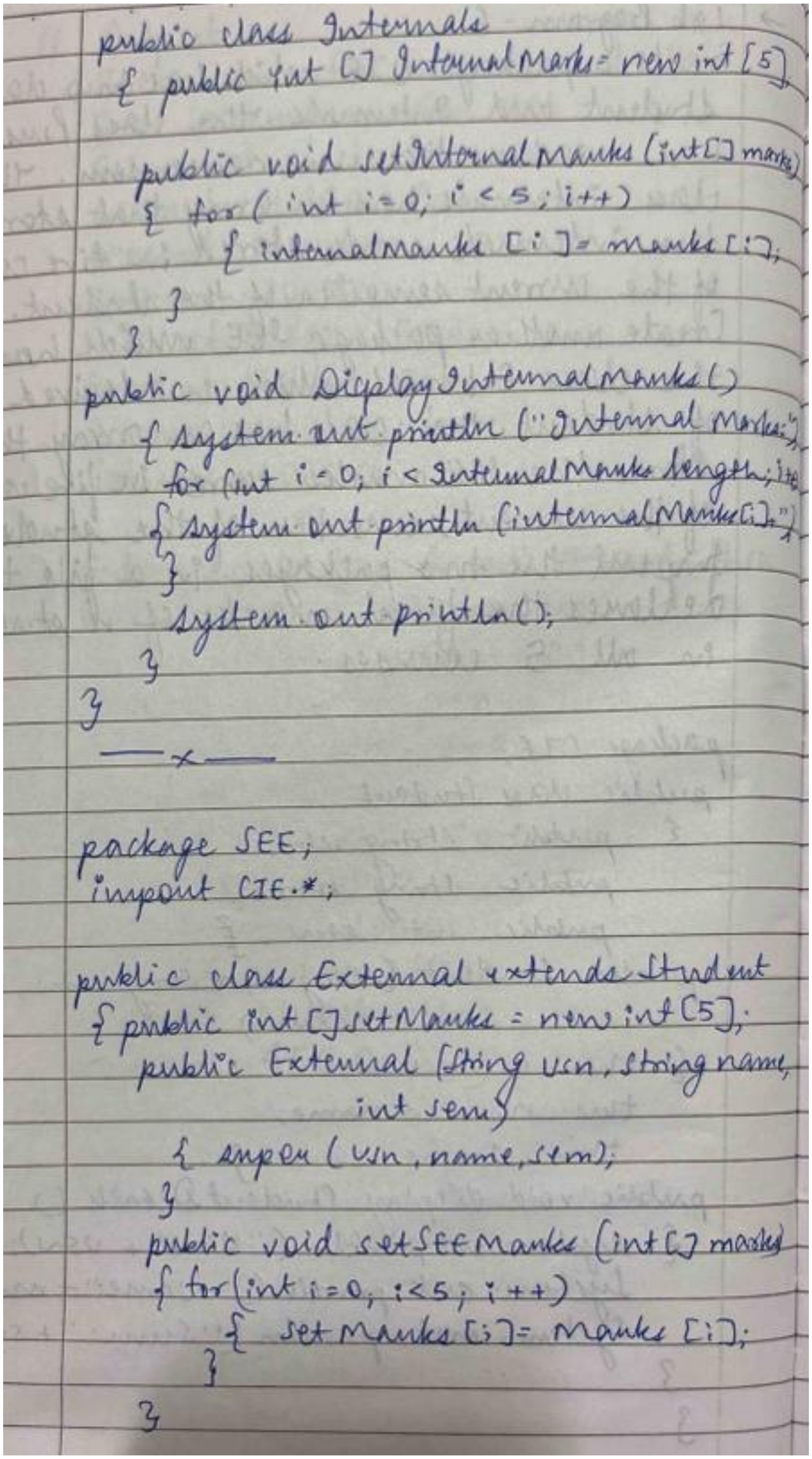
Packages

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

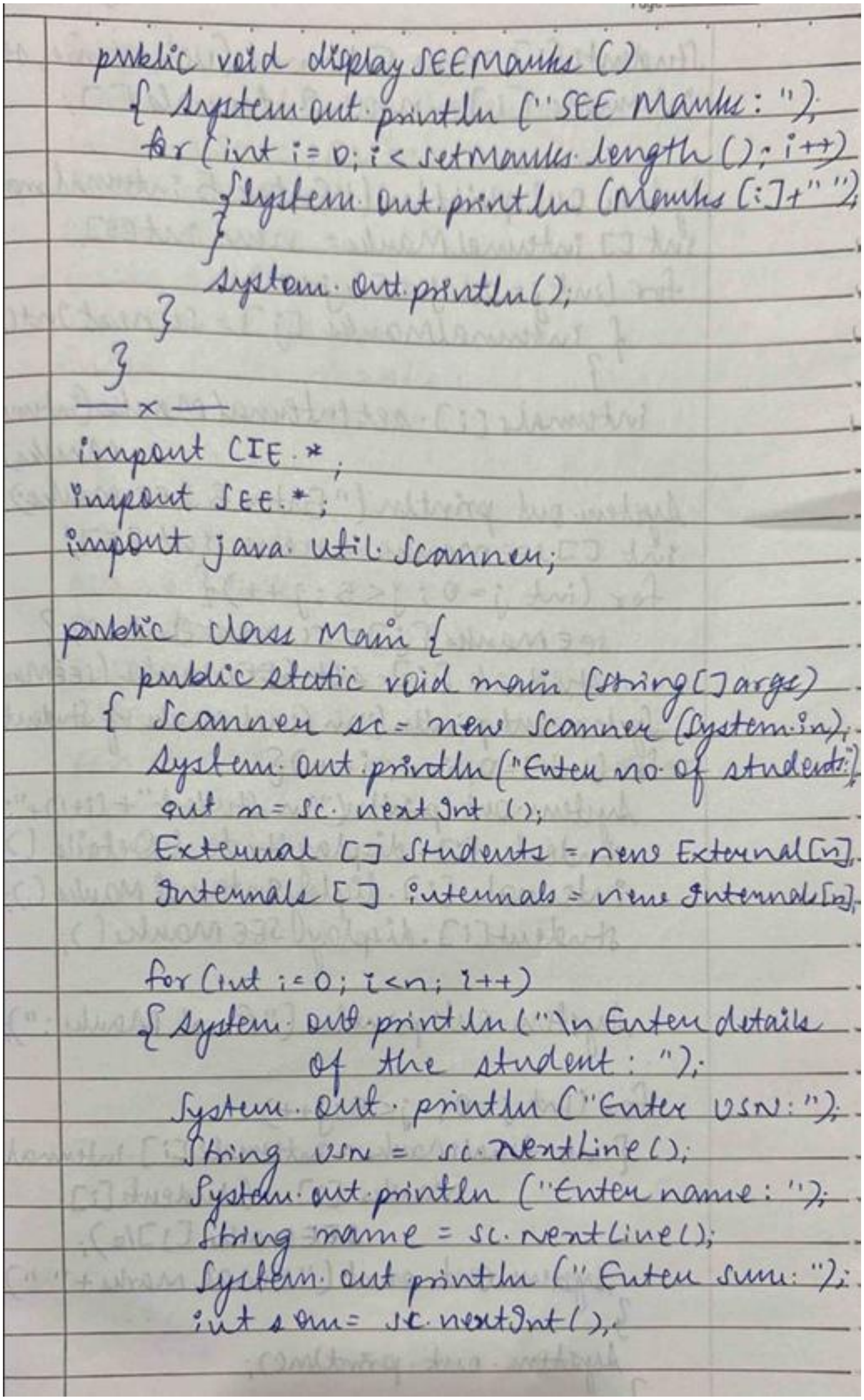
Algorithm:



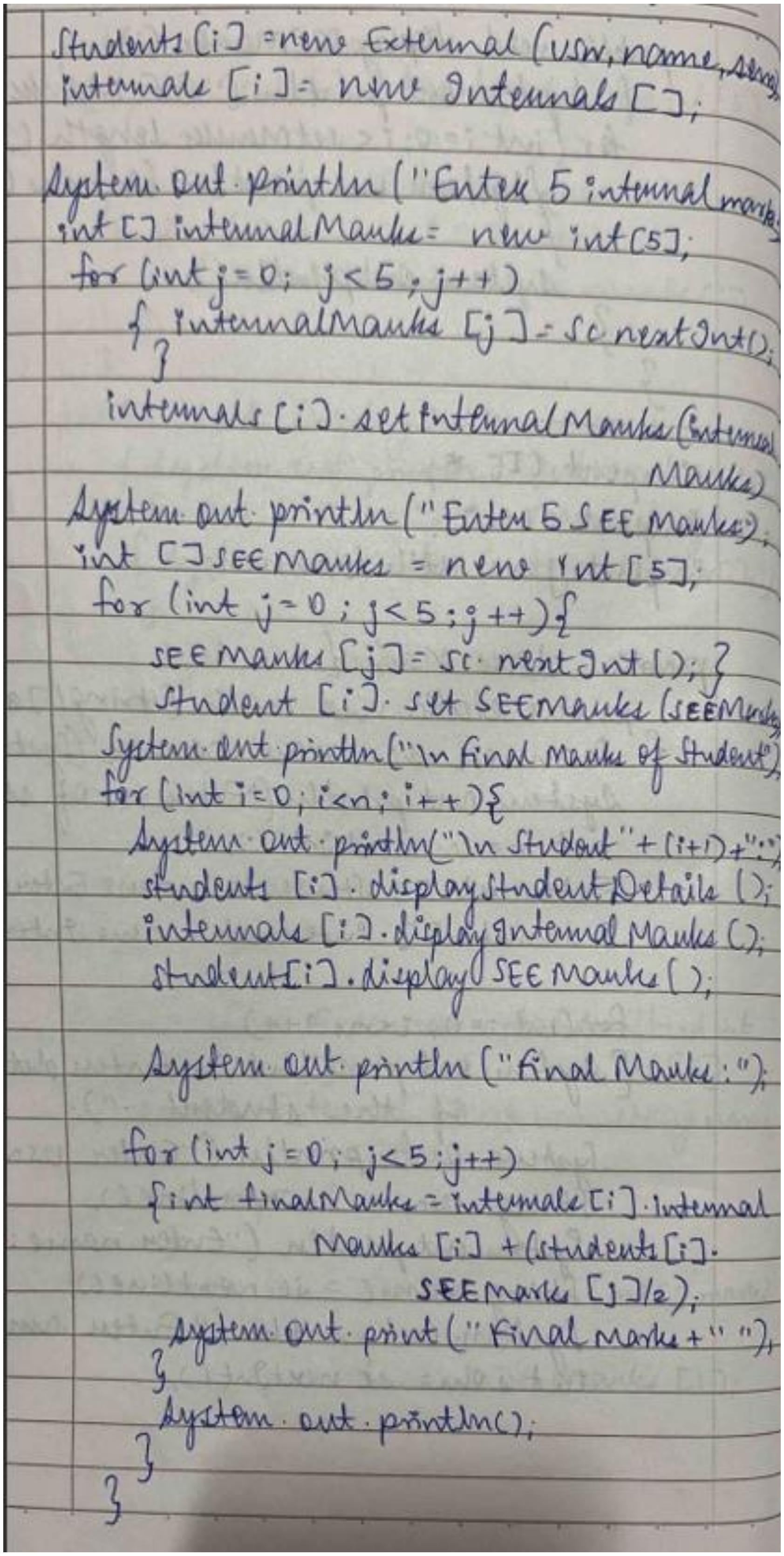
```
package CIE;
public class Student
 { public String usn;
   public String name;
   public int sem; }
public Student (String usn, String name, int
                                        sem)
{ this.usn= usn;
  this.name = name;
  this.sem= sem; }
public void display StudentDetails ()
{ System.out.println ("USN:" + usn);
  System.out.println(" Name:" + name);
  System.out.println ("Sum:" + sum);
 }
}
```



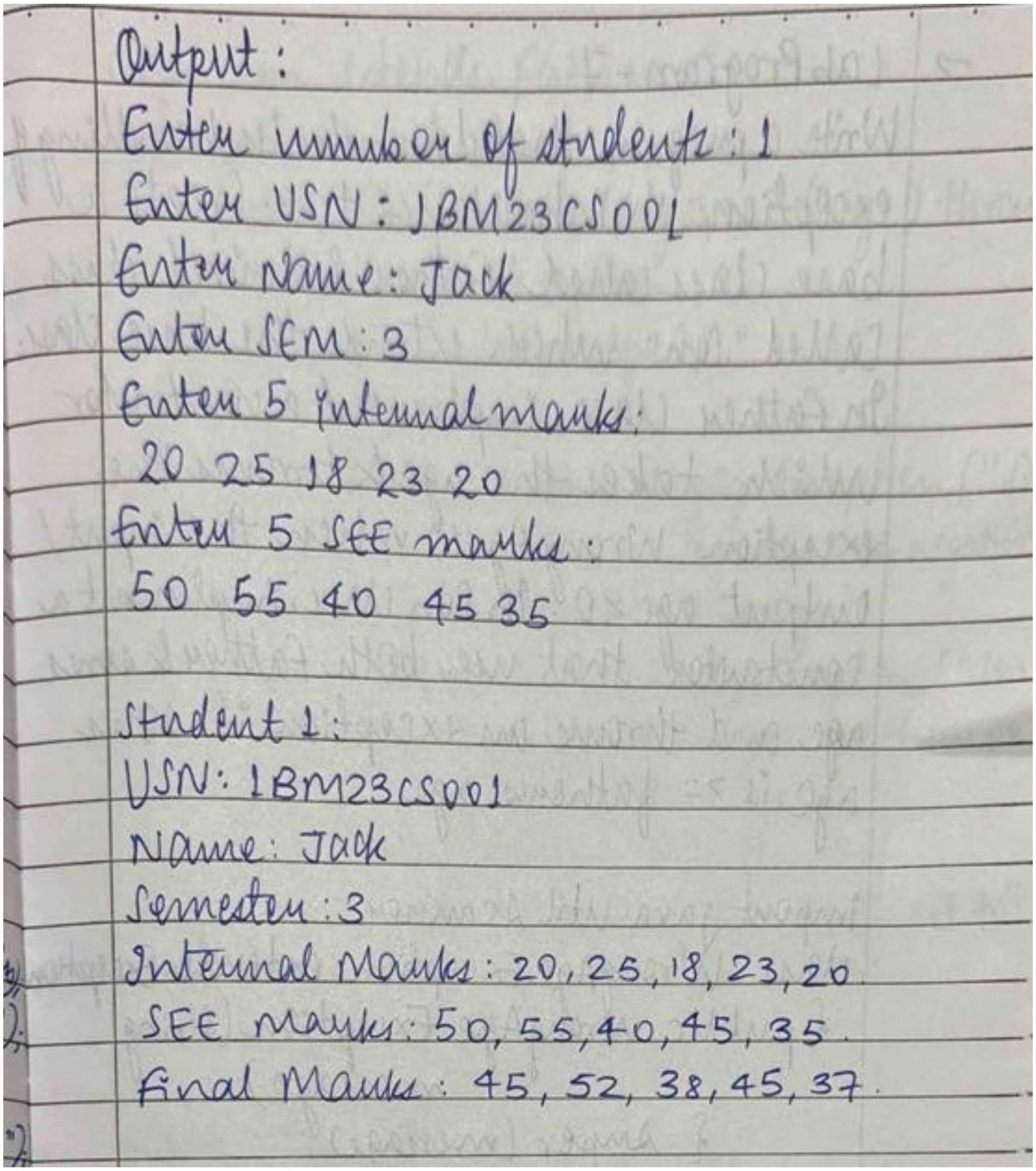
```
public class Internals
 { public int [] InternalMarks= new int [5];

   public void setInternalMarks (int[] marks)
   { for ( int i=0; i < 5; i++)
        { internalMarks [i]= marks [i];
      }
   }
public void DisplayInternalMarks()
   { System.out.println ("Internal Marks:");
    for (int i = 0; i < InternalMarks.length; i++)
    { System.out.println (internalMarks[i]+" ");
    }
     System.out.println();
   }
}
  ——x——

package SEE;
import CIE.*;

public class External extends Student
 { public int []setMarks = new int [5];
   public External (String usn, String name,
                 int sem)
     { super (usn, name, sem);
     }
   public void setSEEMarks (int[] marks)
   { for(int i=0, i<5; i++)
        { setMarks [i]= marks [i];
      }
   }
```



```
public void display SEEMarks ()
 { System.out.println ("SEE Marks: ");
  for (int i=0; i< setMarks.length (); i++)
     {System.out.println (Marks [i]+" ");
     }
     System.out.println();
  }
 }
 —×—
import CIE.*;
import SEE.*;
import java.util.Scanner;

public class Main {
  public static void main (String[] args)
  { Scanner sc = new Scanner (System.in);
    System.out.println("Enter no. of students");
    int n = sc.nextInt ();
    External [] Students = new External[n];
    Internals [] internals = new Internals[n];

    for (int i=0; i<n; i++)
    { System.out.println ("\n Enter details
                 of the student: ");
      System.out.println ("Enter USN: ");
      String usn = sc.nextLine();
      System.out.println ("Enter name: ");
      String name = sc.nextLine();
      System.out.println (" Enter sum: ");
      int sum = sc.nextInt();
```



```
Students [i] = new External (usn, name, sem
internals [i] = new Internals [];

System.out.println ("Enter 5 internal marks
int [] internalMarks = new int [5];
 for (int j=0; j<5; j++)
     { internalMarks [j] = sc.nextInt();
      }
   internals [i].setInternalMarks (internal
                                   Marks)
 System.out.println ("Enter 5 SEE Marks:");
 int [] SEEMarks = new int [5];
  for (int j=0; j<5; j++){
       SEEMarks [j] = sc.nextInt(); }
       Student [i].set SEEMarks (SEEMarks
   System.out.println ("\n Final Marks of Student")
   for (int i=0; i<n; i++){
    System.out.println ("\n Student" + (i+1) + ":")
     students [i].displayStudentDetails ();
     internals [i].displayInternalMarks ();
     student[i].display SEEMarks ();

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

    for (int j=0; j<5; j++)
     {int finalMarks = internals [i].internal
                    Marks [i] + (students [i].
                           SEEMarks [j]/2);
       System.out.print ("Final marks + " ");
     }
      System.out.println();
    }
 }
```



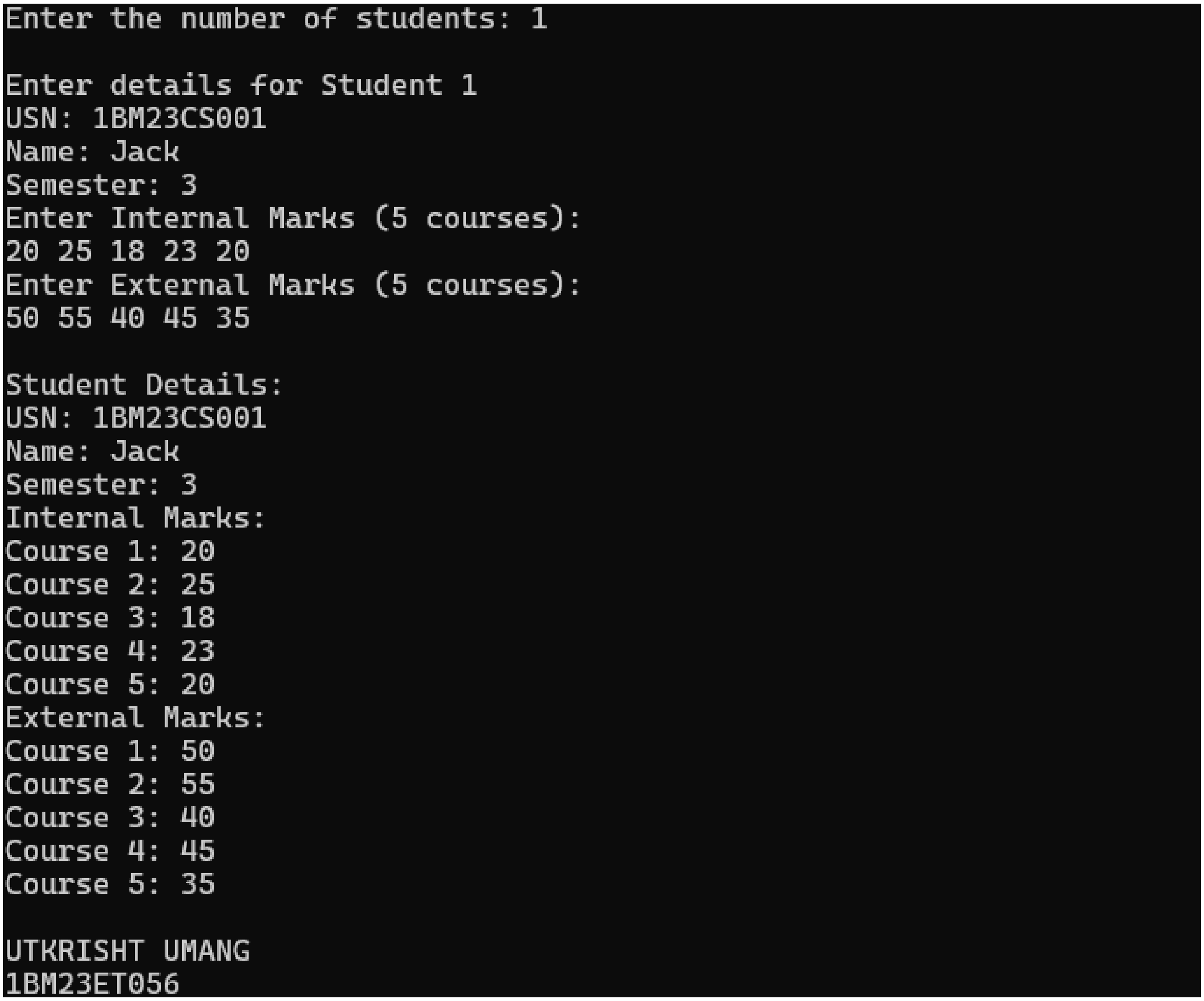
Output :

```
Enter number of students : 1
Enter USN : 1BM23CS001
Enter Name : Jack
Enter SEM : 3
Enter 5 internal marks :
 20 25 18 23 20
Enter 5 SEE marks :
 50 55 40 45 35

Student 1 :
USN : 1BM23CS001
Name : Jack
Semester : 3
Internal Marks : 20, 25, 18, 23, 20.
 SEE Marks : 50, 55, 40, 45, 35.
 Final Marks : 45, 52, 38, 45, 37.
```

Code:

```
package CIE;

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

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

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

package CIE;

public class Internals extends Student {
    private 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 i = 0; i < internalMarks.length; i++) {
            System.out.println("Course " + (i + 1) + ": " + internalMarks[i]);
        }
    }
}

package SEE;

import CIE.Student;

public class External extends Student {
    private 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 i = 0; i < externalMarks.length; i++) {
            System.out.println("Course " + (i + 1) + ": " + externalMarks[i]);
        }
    }
}

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

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

        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();
        sc.nextLine(); // Consume newline

        Internals[] internals = new Internals[n];
        External[] externals = new External[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Student " + (i + 1));
            System.out.print("USN: ");
            String usn = sc.nextLine();
            System.out.print("Name: ");
            String name = sc.nextLine();
            System.out.print("Semester: ");
            int sem = sc.nextInt();
            sc.nextLine(); // Consume newline

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

            internals[i] = new Internals(usn, name, sem, internalMarks);

            System.out.println("Enter External Marks (5 courses): ");
            int[] externalMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                externalMarks[j] = sc.nextInt();
            }
            sc.nextLine(); // Consume newline

            externals[i] = new External(usn, name, sem, externalMarks);
        }

        System.out.println("\nStudent Details:");
        for (int i = 0; i < n; i++) {
            internals[i].displayPersonalDetails();
            internals[i].displayInternalMarks();
            externals[i].displayExternalMarks();
            System.out.println();
        }

        System.out.println("UTKRISHT UMANG\n1BM23ET056");

        sc.close();
    }
}
```

Output:

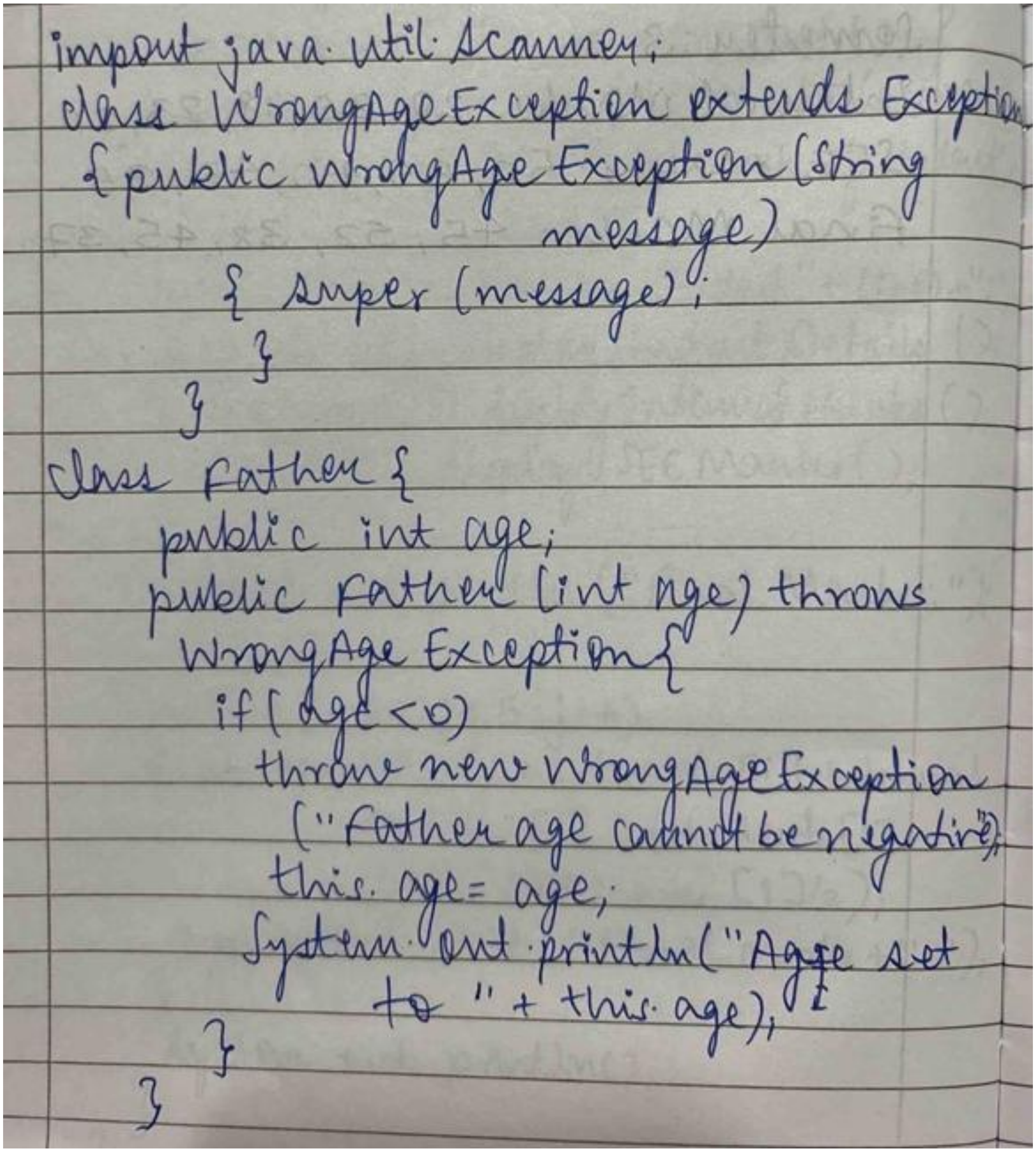


```
Enter the number of students: 1

Enter details for Student 1
USN: 1BM23CS001
Name: Jack
Semester: 3
Enter Internal Marks (5 courses):
20 25 18 23 20
Enter External Marks (5 courses):
50 55 40 45 35

Student Details:
USN: 1BM23CS001
Name: Jack
Semester: 3
Internal Marks:
Course 1: 20
Course 2: 25
Course 3: 18
Course 4: 23
Course 5: 20
External Marks:
Course 1: 50
Course 2: 55
Course 3: 40
Course 4: 45
Course 5: 35

UTKRISHT UMANG
1BM23ET056
```

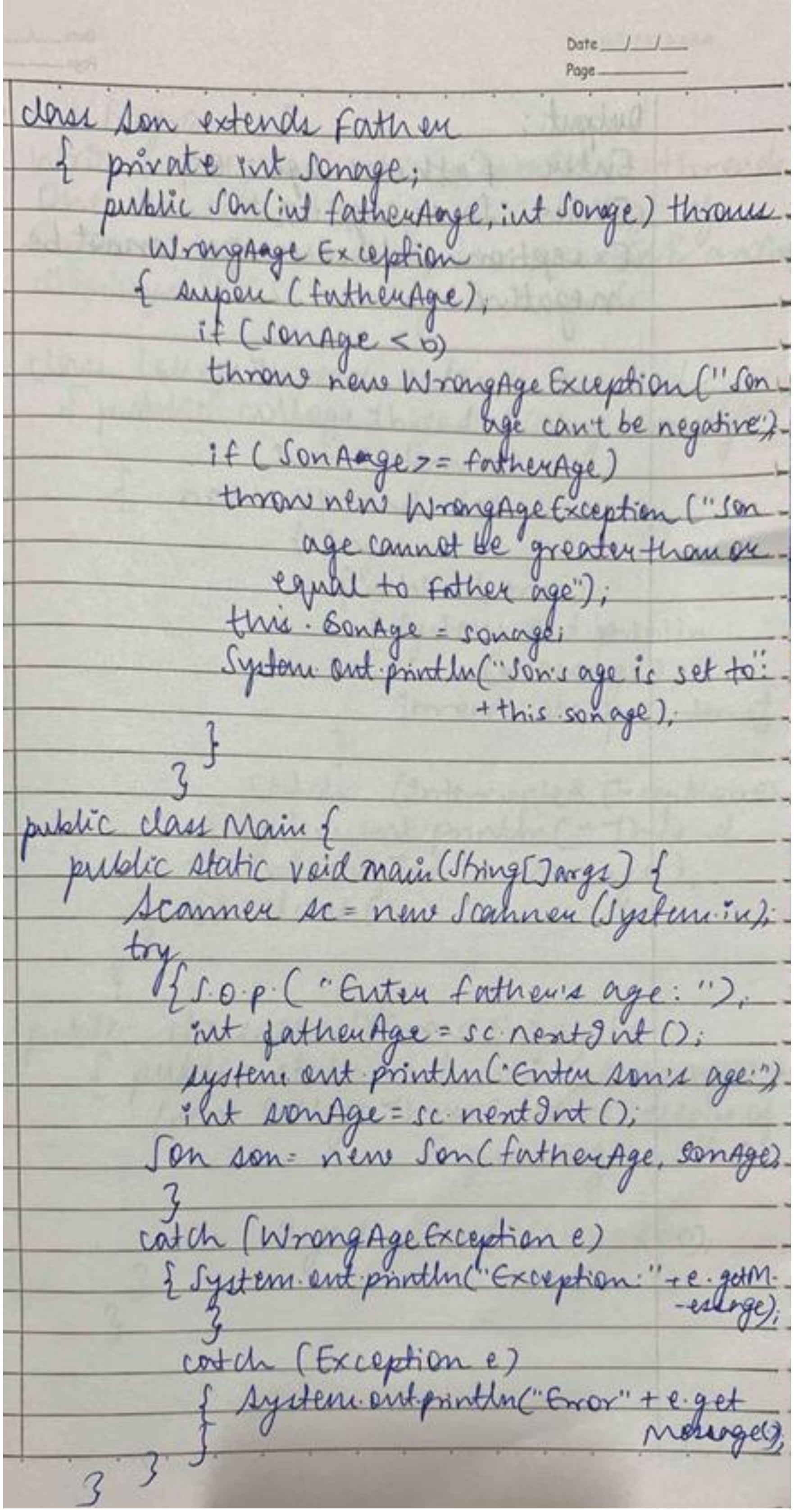
**Program 7**

Exception Handling
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that uses both father and son's age and throws an exception if son's age is >=father’s age.

Algorithm:



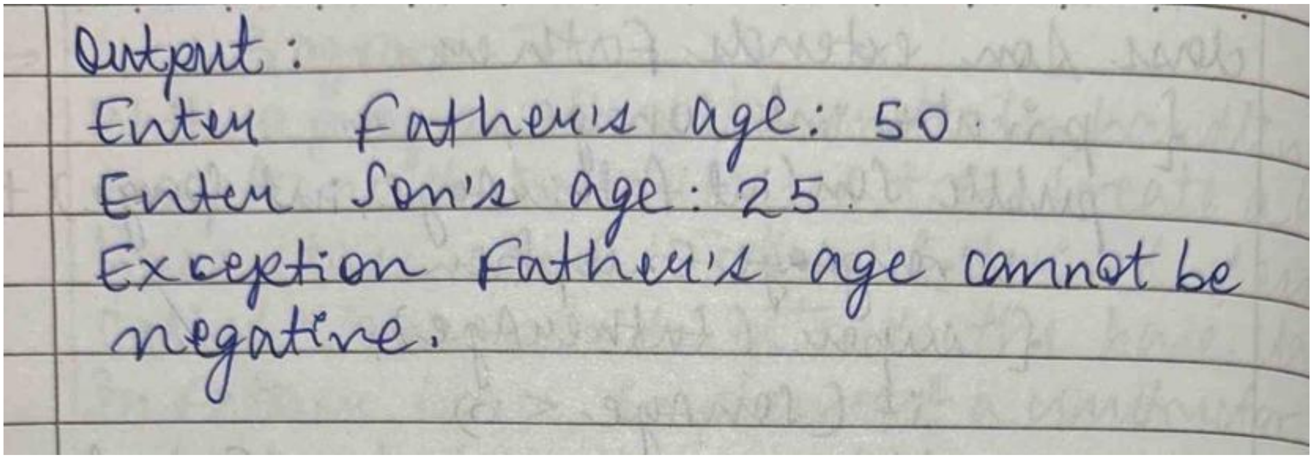
```
import java.util.Scanner;
class WrongAgeException extends Exception
 {public WrongAgeException(String
                        message)
      { super(message);
       }
   }
class Father{
    public int age;
    public Father(int age) throws
     WrongAgeException{
      if(age<0)
        throw new WrongAgeException
          ("Father age cannot be negative");
        this.age= age;
       System.out.println("Agge set
              to " + this.age);
     }
  }
```



Date __/__/__

Page ______

```
class Son extends Father
 { private int sonage;
   public Son(int fatherAge, int Sonage) throws
     WrongAage Exception
    { super (fatherAge);
       if (sonAge < 0)
       throw new WrongAge Exception ("Son
                        age can't be negative");
       if ( SonAage >= fatherAge)
       throw new WrongAge Exception ("Son
              age cannot be greater than or
            equal to father age");
        this . SonAge = sonage;
        System.out.println("Son's age is set to:
                        + this.sonage);
      }
   }
public class Main {
  public static void main(String[]args) {
     Scanner sc = new Scanner (System.in);
     try
      { S.o.p ("Enter father's age: ");
       int fatherAge = sc.nextInt ();
       system.out.println("Enter son's age:");
       int sonAge = sc.nextInt ();
      Son son= new Son(fatherAge, sonAge);
      }
     catch (WrongAge Exception e)
      { System.out.println("Exception:" + e.getM-
                                         -essage);
       }
      catch (Exception e)
       { System.out.println("Error" + e.get
                                   Message();
       }
   }
 }
```



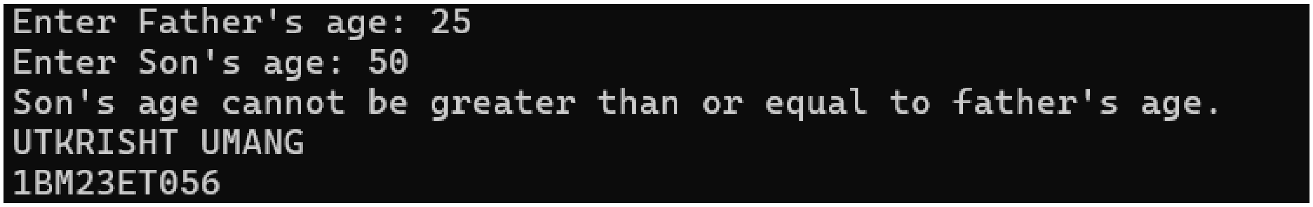
Output :

```
Enter Father's age: 50
Enter Son's age: 25
Exception Father's age cannot be
negative.
```

Code:

```
import java.util.Scanner;

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

class Father {
    int age;

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

class Son extends Father {
    int sonAge;

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

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

        try {
            System.out.print("Enter Father's age: ");
            int fatherAge = sc.nextInt();

            System.out.print("Enter Son's age: ");
            int sonAge = sc.nextInt();

            Father father = new Father(fatherAge);
            Son son = new Son(fatherAge, sonAge);

            System.out.println("Father's age: " + father.age);
            System.out.println("Son's age: " + son.sonAge);

        } catch (WrongAgeException e) {
            System.out.println(e.getMessage());
        }

        System.out.print("UTKRISHT UMANG\n1BM23ET056");

        sc.close();
    }
}
```

Output:



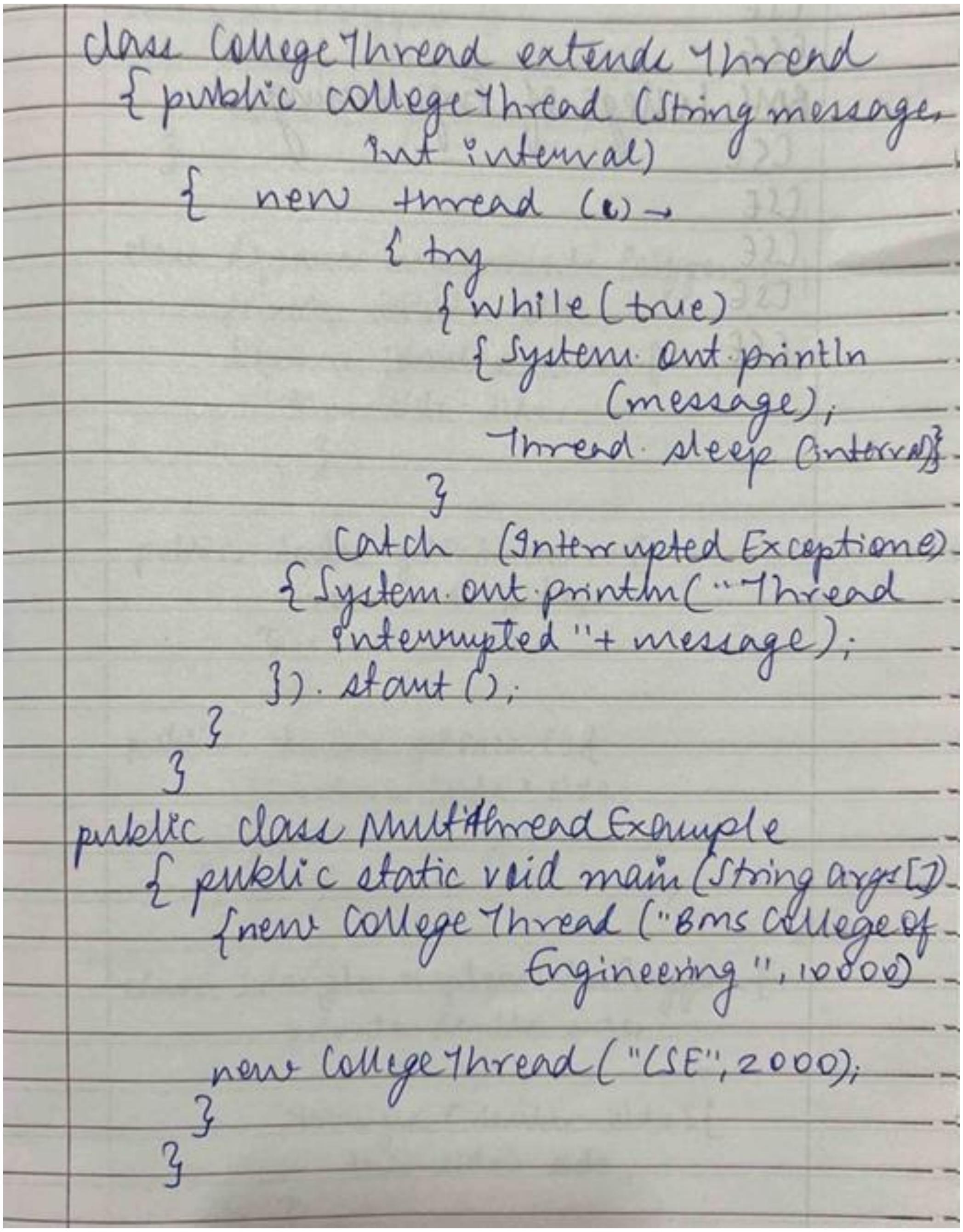
```
Enter Father's age: 25
Enter Son's age: 50
Son's age cannot be greater than or equal to father's age.
UTKRISHT UMANG
1BM23ET056
```

**Program 8**

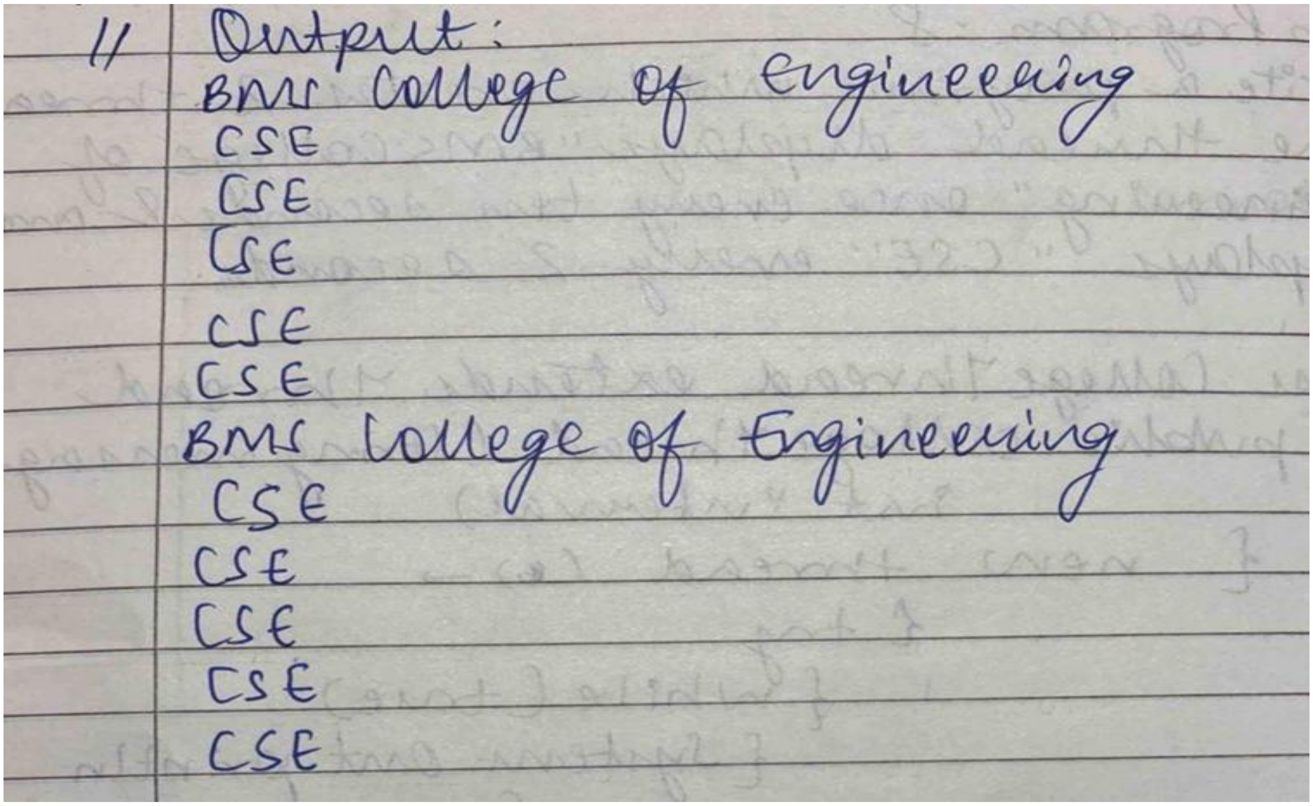
Threads

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:



```
class CollegeThread extends Thread
  {public collegeThread (String message,
              Int interval)
    {  new thread (()→
              { try
                  {while (true)
                     {System.out.println
                              (message);
                      Thread.sleep (interval)}
                  }
              catch (Interrupted Exception e)
          {System.out.println(" Thread
              interrupted "+ message);
          }).start ();
      }
   }
public class MultiThreadExample
   { public static void main(String args[])
     {new CollegeThread ("BMS College of
                      Engineering", 10000)

     new CollegeThread ("CSE", 2000);
     }
   }
```

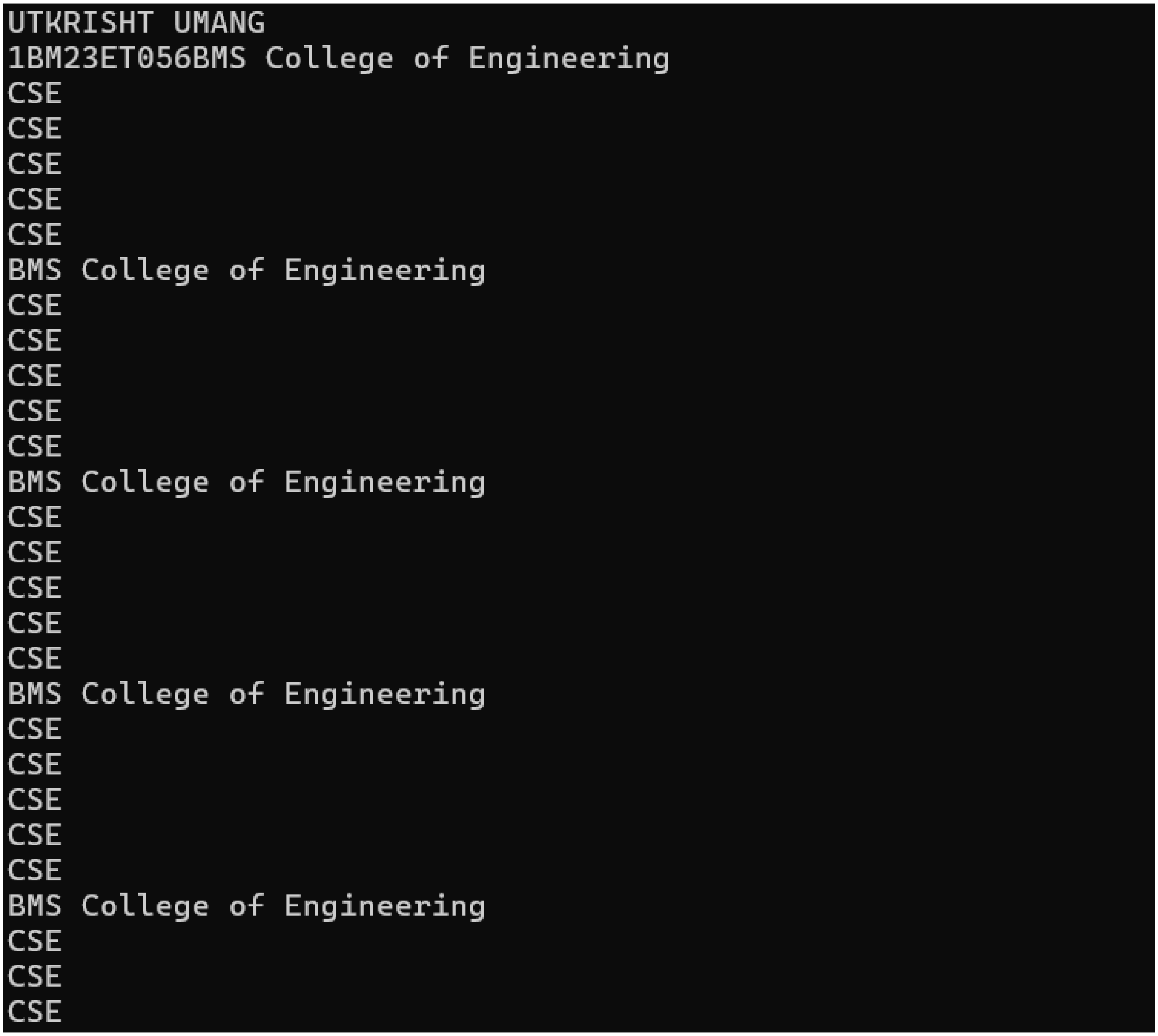


```
// Output:
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```

Code:

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

    public MessageThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }

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

public class MultiThreadExample {
    public static void main(String[] args) {
        // Create two threads with different messages and intervals
        MessageThread thread1 = new MessageThread("BMS College of Engineering", 10000); // 10 seconds
        MessageThread thread2 = new MessageThread("CSE", 2000); // 2 seconds

        // Start both threads
        thread1.start();
        thread2.start();

        // Print the tag
        System.out.print("UTKRISHT UMANG\n1BM23ET056");
    }
}
```

Output:

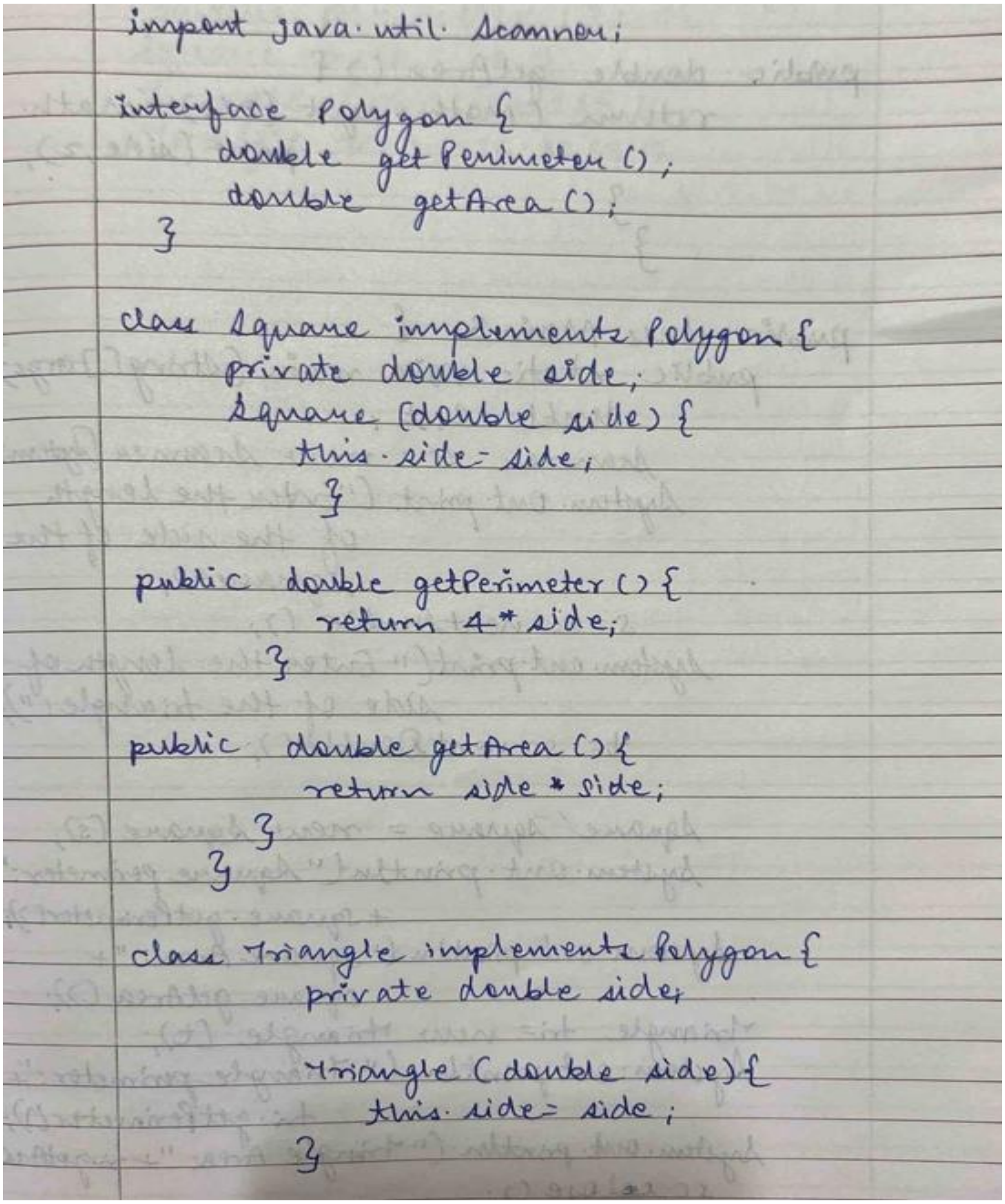


```
UTKRISHT UMANG
1BM23ET056BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
```

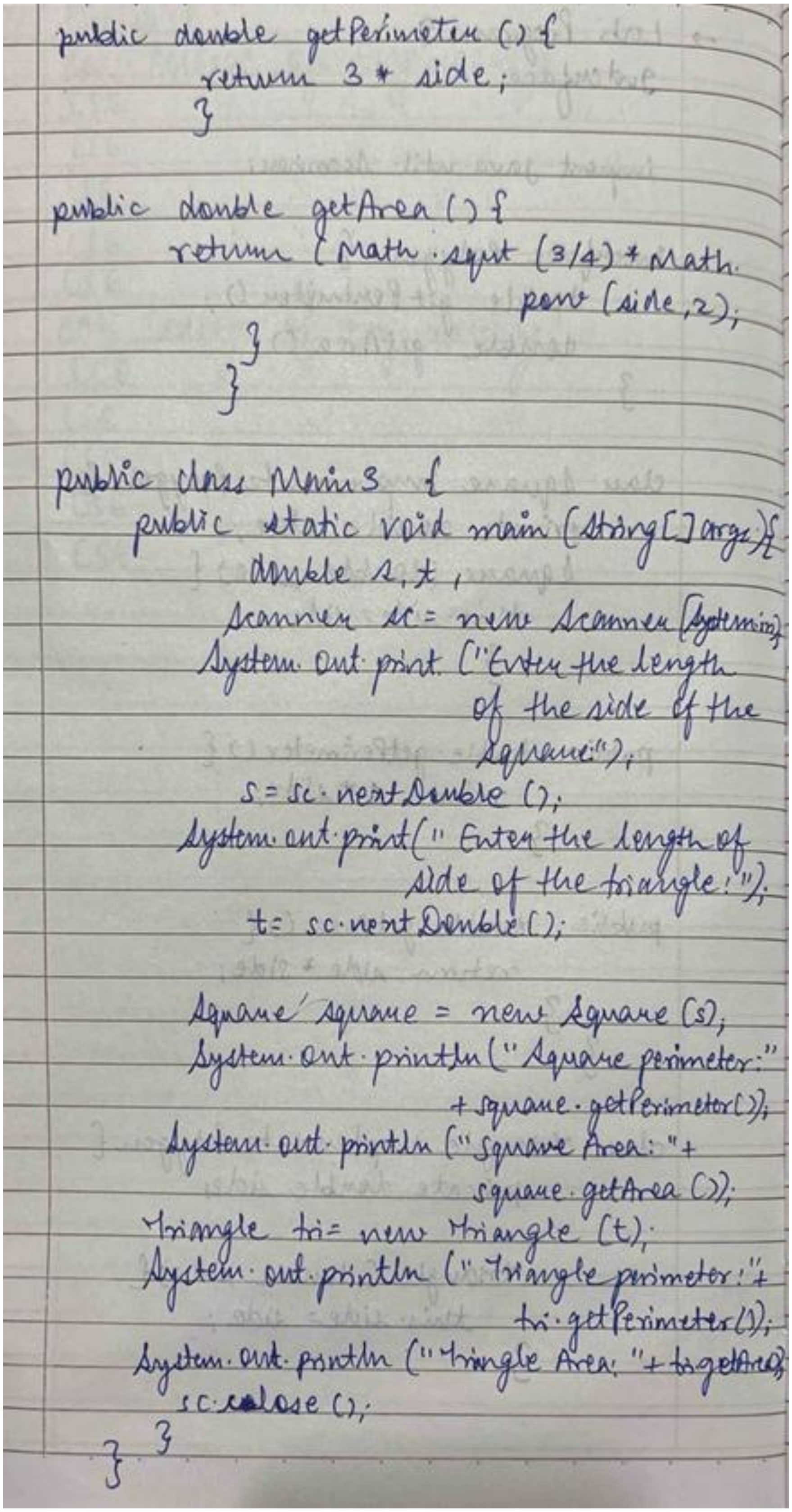
**Program 7**

Interfaces

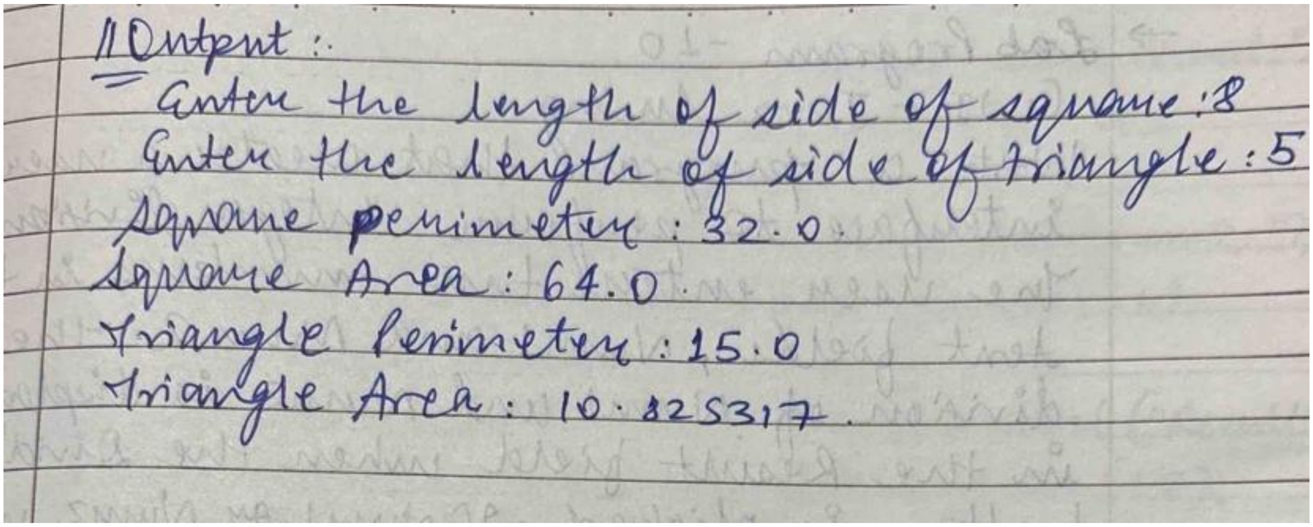
Algorithm:



```
import java.util.Scanner;

interface Polygon {
        double get Perimeter ();
        double   getArea ();
  }

class Square implements Polygon {
        private double side;
        Square (double side) {
            this.side= side;
             }

 public  double getPerimeter () {
             return 4* side;
         }

public  double getArea () {
            return side * side;
       }
   }

class Triangle implements Polygon {
            private double side;

            Triangle (double side){
                this.side= side;
            }
```



```
public double getPerimeter (){
        return 3 * side;
        }

public double getArea (){
       return (Math.sqrt (3/4) * Math.
                          pow (side,2);
          }
      }

public class Main3 {
    public static void main (String[] args){
           double s,t,
          Scanner sc = new Scanner (System.in);
        System.out.print ("Enter the length
                     of the side of the
                     Square:");
          s = sc.nextDouble ();
       System.out.print(" Enter the length of
                      side of the triangle:");
          t= sc.nextDouble();

       Square square = new Square (s);
       System.out.println("Square perimeter:"
                   + square.getPerimeter());
      System.out.println ("Square Area: "+
                     square.getArea ());
     Triangle tri= new Triangle (t);
     System.out.println (" Triangle perimeter:"+
                        tri.getPerimeter());
    System.out.println ("Triangle Area: "+ tri.getArea());
       sc.close ();
     }
  }
```

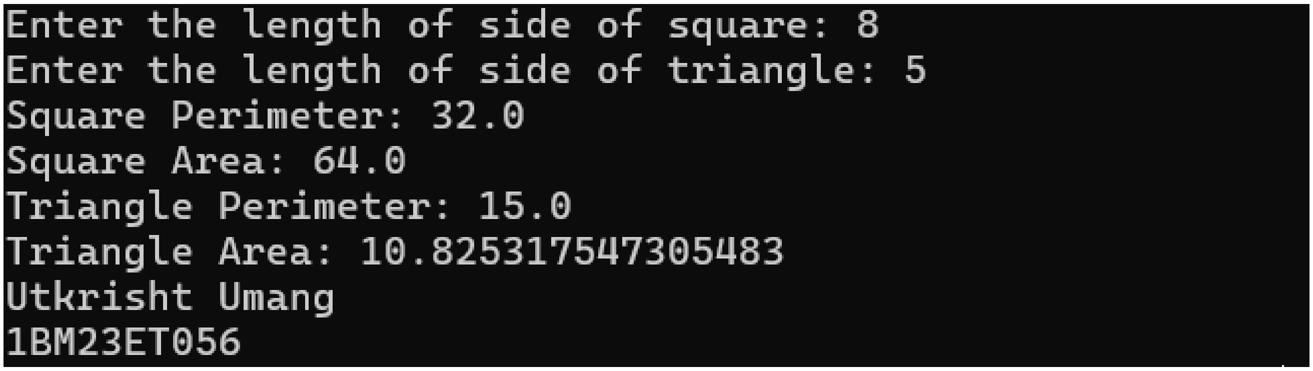


```
//Output :-
Enter the length of side of square:8
Enter the length of side of Triangle:5
Square perimeter : 32.0.
Square Area : 64.0.
Triangle Perimeter : 15.0
Triangle Area : 10.825317.
```

Code:

```
import java.util.Scanner;

interface Polygon {
    double getPerimeter();
    double getArea();
}

class Square implements Polygon {
    private double side;

    Square(double side) {
        this.side = side;
    }

    @Override
    public double getPerimeter() {
        return 4 * side;
    }

    @Override
    public double getArea() {
        return side * side;
    }
}

class Triangle implements Polygon {
    private double side;

    Triangle(double side) {
        this.side = side;
    }
    @Override
    public double getPerimeter() {
        return 3 * side;
    }

    @Override
    public double getArea() {
        return (Math.sqrt(3) / 4) * Math.pow(side, 2);
    }
}

public class maininterface {
    public static void main(String[] args) {
        double s, t;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the length of side of square: ");
        s = sc.nextDouble();

        System.out.print("Enter the length of side of triangle: ");
        t = sc.nextDouble();

        Square square = new Square(s);
        System.out.println("Square Perimeter: " + square.getPerimeter());
        System.out.println("Square Area: " + square.getArea());

        Triangle tri = new Triangle(t);
        System.out.println("Triangle Perimeter: " + tri.getPerimeter());
        System.out.println("Triangle Area: " + tri.getArea());
        System.out.print("Chethan K S\n1BM23CS074");
        sc.close();
    }
}
```

Output:

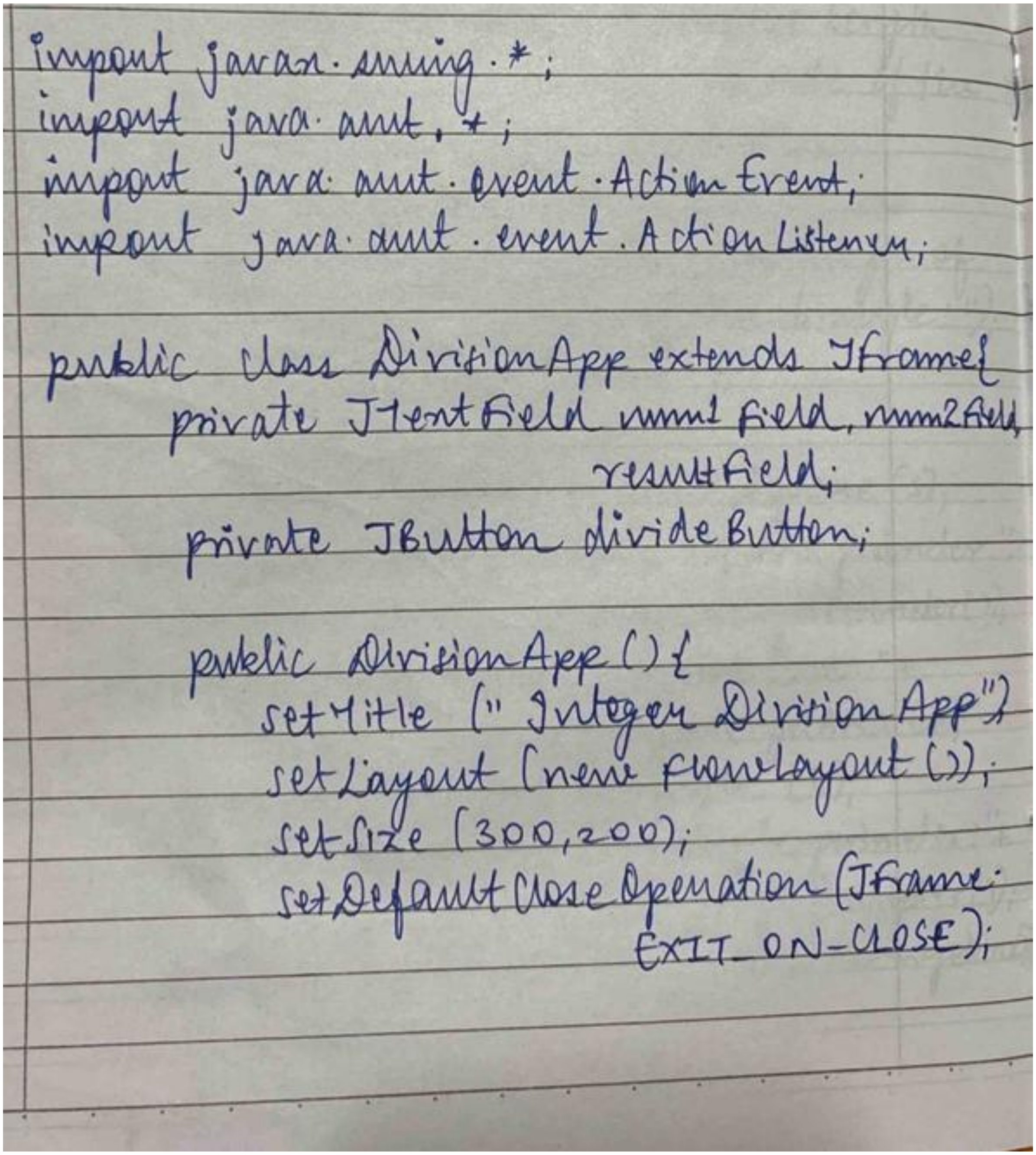


```
Enter the length of side of square: 8
Enter the length of side of triangle: 5
Square Perimeter: 32.0
Square Area: 64.0
Triangle Perimeter: 15.0
Triangle Area: 10.825317547305483
Utkrisht Umang
1BM23ET056
```

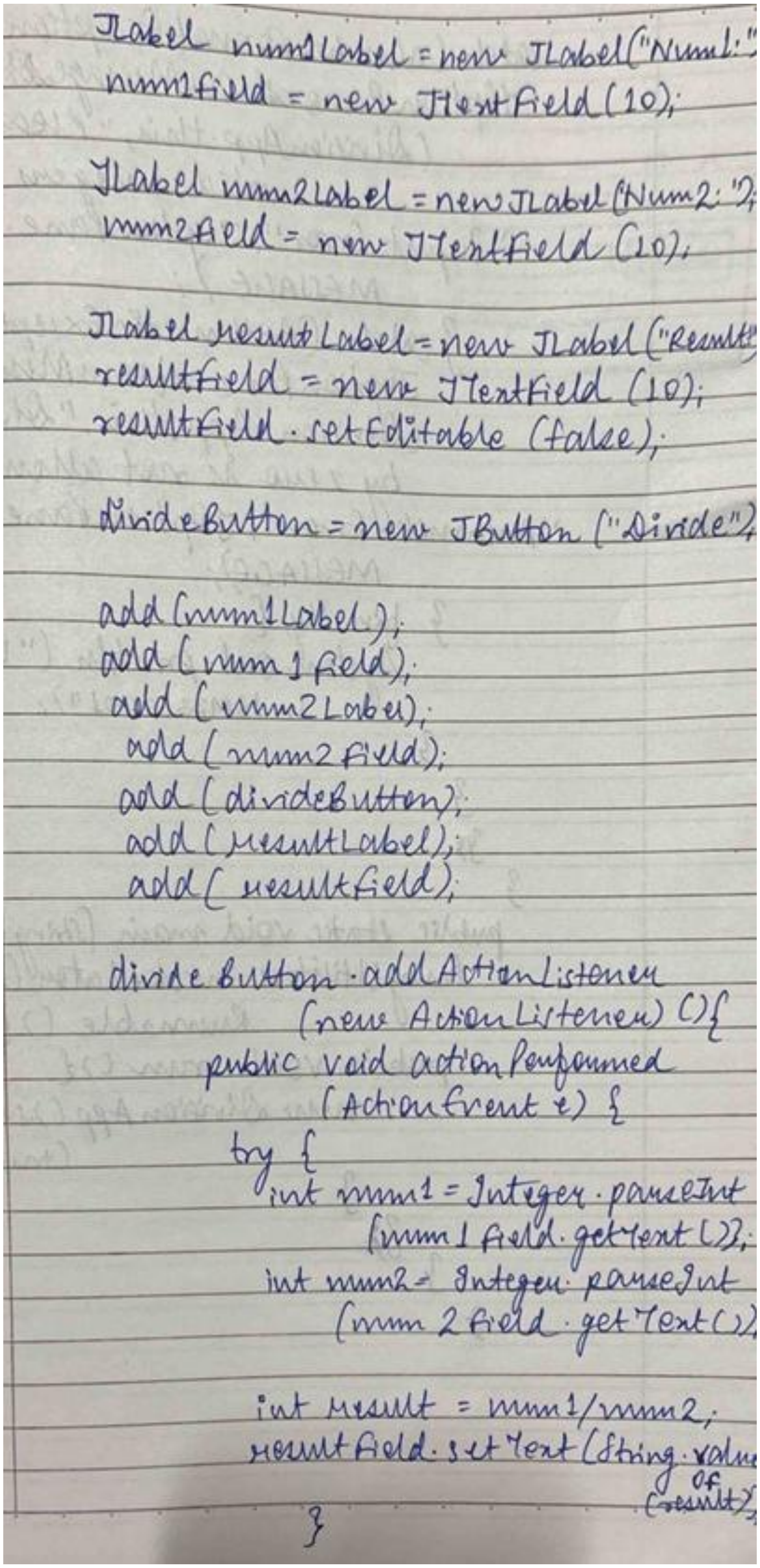
**Program 10**

GUI – Java Swing

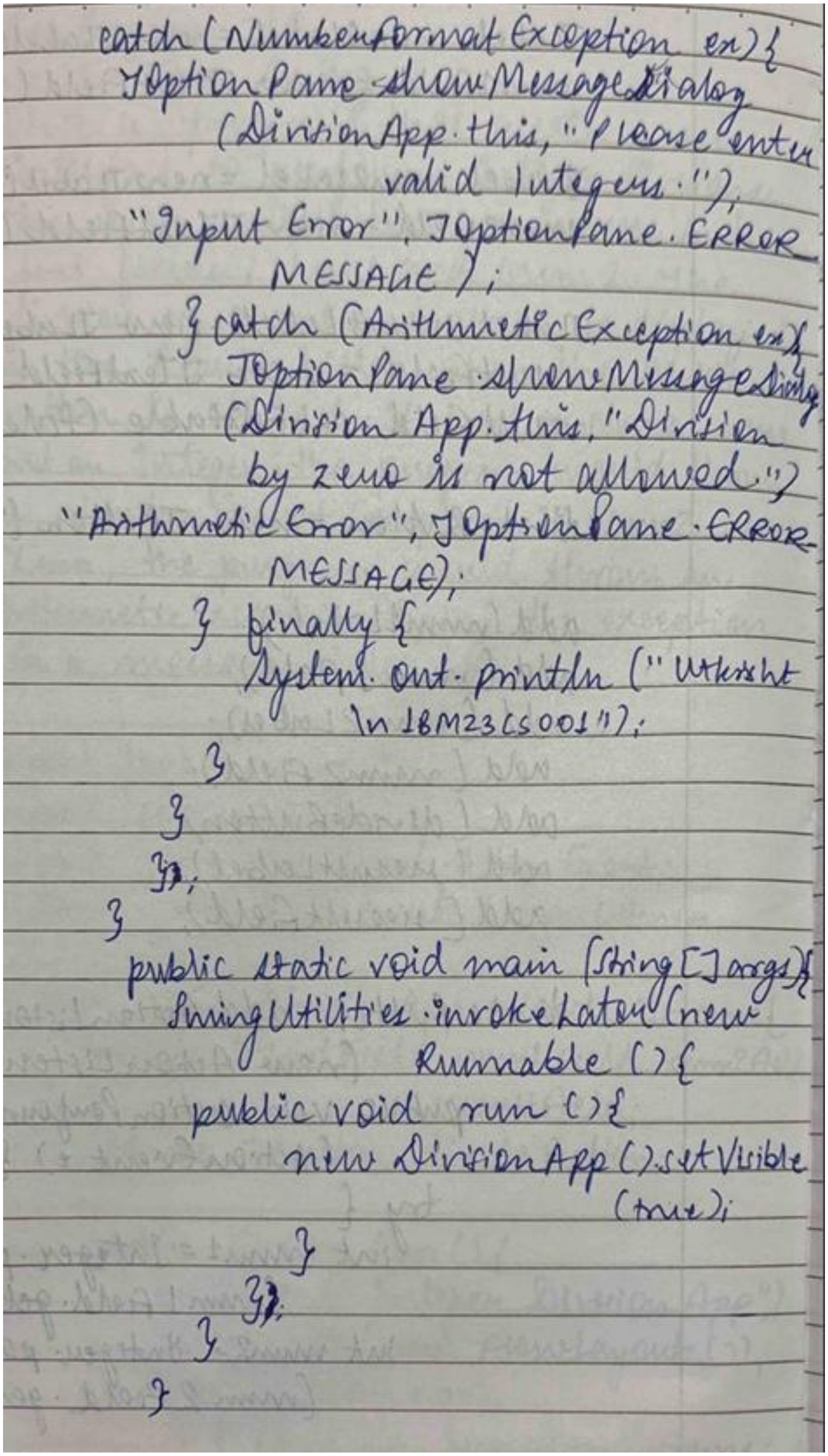
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException.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



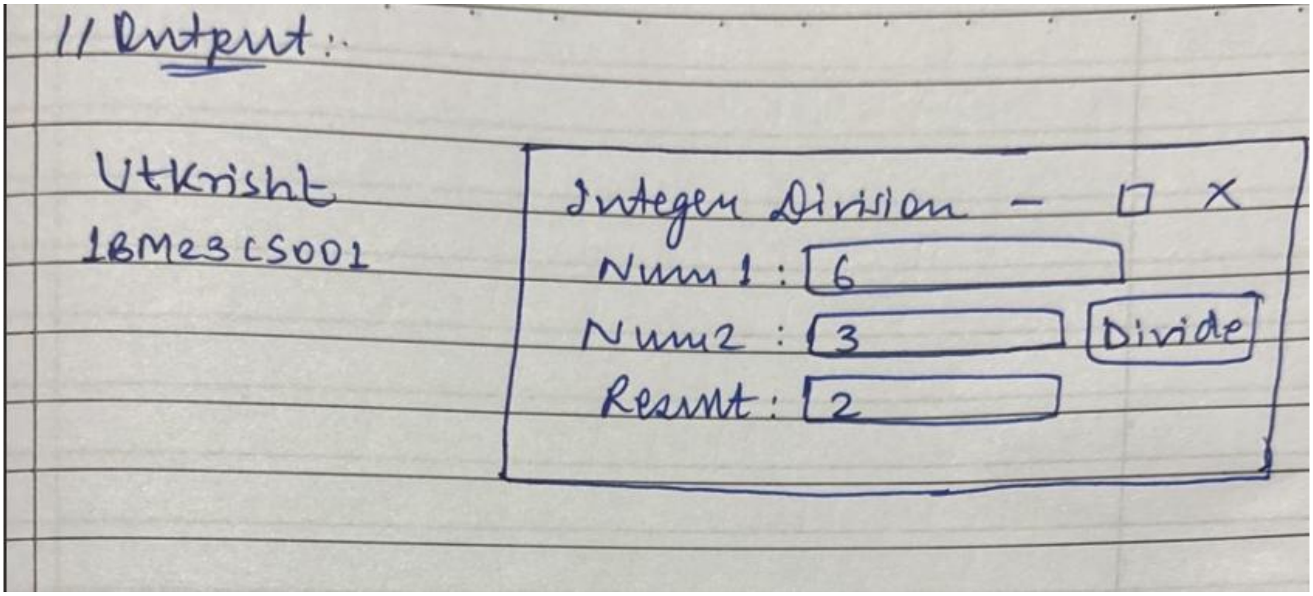
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class DivisionApp extends JFrame{
      private JTextField num1Field, num2Field,
                                resultField;
      private JButton divideButton;

      public DivisionApp(){
         setTitle("Integer Division App")
         setLayout(new FlowLayout());
         setSize(300,200);
         setDefaultCloseOperation(JFrame.
                              EXIT_ON-CLOSE);
```



```
JLabel num1Label = new JLabel("Num1:"
num1field = new JTextField(10);

JLabel num2Label = new JLabel("Num2:");
num2field = new JTextField (10);

JLabel resultLabel = new JLabel ("Result
resultField = new JTextField (10);
resultField.setEditable (false);

divideButton = new JButton ("Divide");

add (num1Label);
add (num1 field);
add (num2Label);
 add (num2 Field);
 add (divideButton);
 add (resultLabel);
 add ( resultField);

divide Button.addActionListener
          (new ActionListener) (){
    public void actionPerformed
             (ActionEvent e) {
        try {
           int num1 = Integer.parseInt
                   (num1 field.getText ());
          int num2 = Integer.parseInt
                (num 2 field.getText());

          int result = num1/num2;
          resultField.setText (String.value
                                      Of
                                  (result));
          }
```



```
catch (NumberFormatException ex){
JOptionPane.showMessageDialog
        (DivisionApp.this, "Please enter
            valid Integers.");
"Input Error", JOptionPane.ERROR
        MESSAGE);
    }catch (ArithmeticException ex){
      JOptionPane.showMessageDialog
      (Division App.this, "Division
       by zero is not allowed.")
"Arithmetic Error", JOptionPane.ERROR
        MESSAGE);
    } finally {
      System.out.println ("Utkrsht
          \n 1BM23CS001");
    }
  }
  });
}
public static void main (String[] args){
  SwingUtilities.invokeLater(new
              Runnable (){
  public void run (){
     new DivisionApp().setVisible
                     (true);
     }
   });
 }
}
```



// Output:

Utkrisht

1BM23CS001

Integer Division − □ X

Num 1: 6

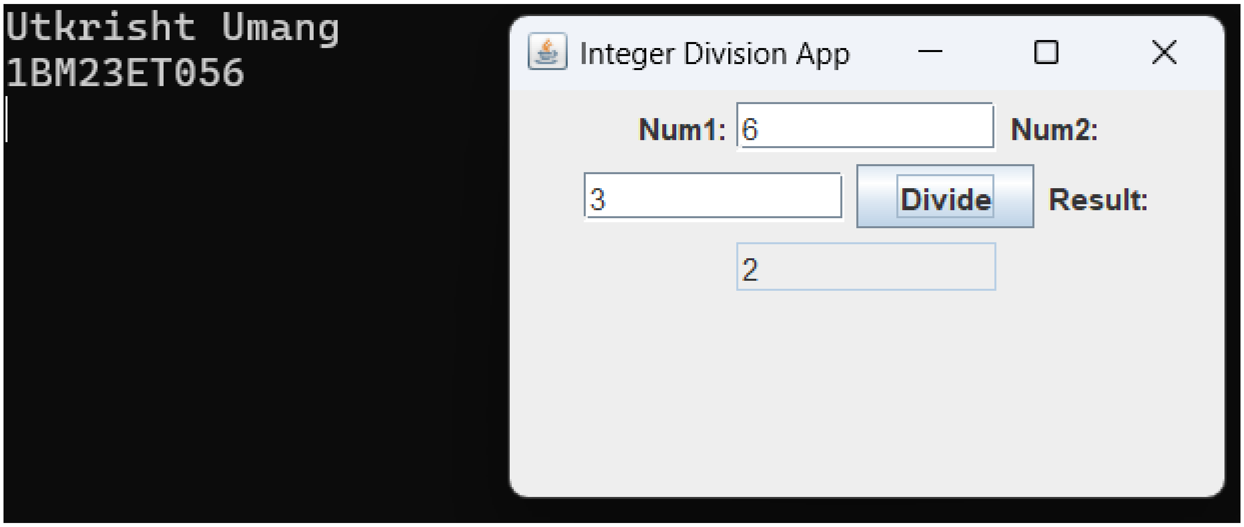
Num2 : 3

Result: 2

Code:

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

public class DivisionApp extends JFrame {
    private JTextField num1Field, num2Field, resultField;
    private JButton divideButton;

    public DivisionApp() {
        setTitle("Integer Division App");
        setLayout(new FlowLayout());
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel num1Label = new JLabel("Num1:");
        num1Field = new JTextField(10);

        JLabel num2Label = new JLabel("Num2:");
        num2Field = new JTextField(10);

        JLabel resultLabel = new JLabel("Result:");
        resultField = new JTextField(10);
        resultField.setEditable(false);

        divideButton = new JButton("Divide");

        add(num1Label);
        add(num1Field);
        add(num2Label);
        add(num2Field);
        add(divideButton);
        add(resultLabel);
        add(resultField);

        divideButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    int num1 = Integer.parseInt(num1Field.getText());
                    int num2 = Integer.parseInt(num2Field.getText());

                    int result = num1 / num2;
                    resultField.setText(String.valueOf(result));

                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(DivisionApp.this, "Please enter valid integers.",
"Input Error", JOptionPane.ERROR_MESSAGE);
                } catch (ArithmeticException ex) {
                    JOptionPane.showMessageDialog(DivisionApp.this, "Division by zero is not allowed.",
"Arithmetic Error", JOptionPane.ERROR_MESSAGE);
                } finally {
                    System.out.println("Utkrisht Umang\n1BM23ET056");
                }
            }
        });
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new DivisionApp().setVisible(true);
            }
        });
    }
}
```

Output:



```
Utkrisht Umang
1BM23ET056
```

Num1: 6

Num2: 3

Result: 2