VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

StudentName (1BM23CS063)

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 **Bhavya Goyal (1BM23CS063),** 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.

Lab faculty Incharge Name Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Sl. No.	Date	Experiment Title	Page No.
1	1/10/2024	Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2 -4ac is negative, display a message stating that there are no real solutions	
2	8/10/2024	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.	
3	15/10/2024	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.	
4	22/10/2024	Create an abstract class animal with method eat and sleep and create three subclass lion, bear, tiger which extends animal class and implement eat and sleep method based on behaviour.	
5	22/10/2024	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	

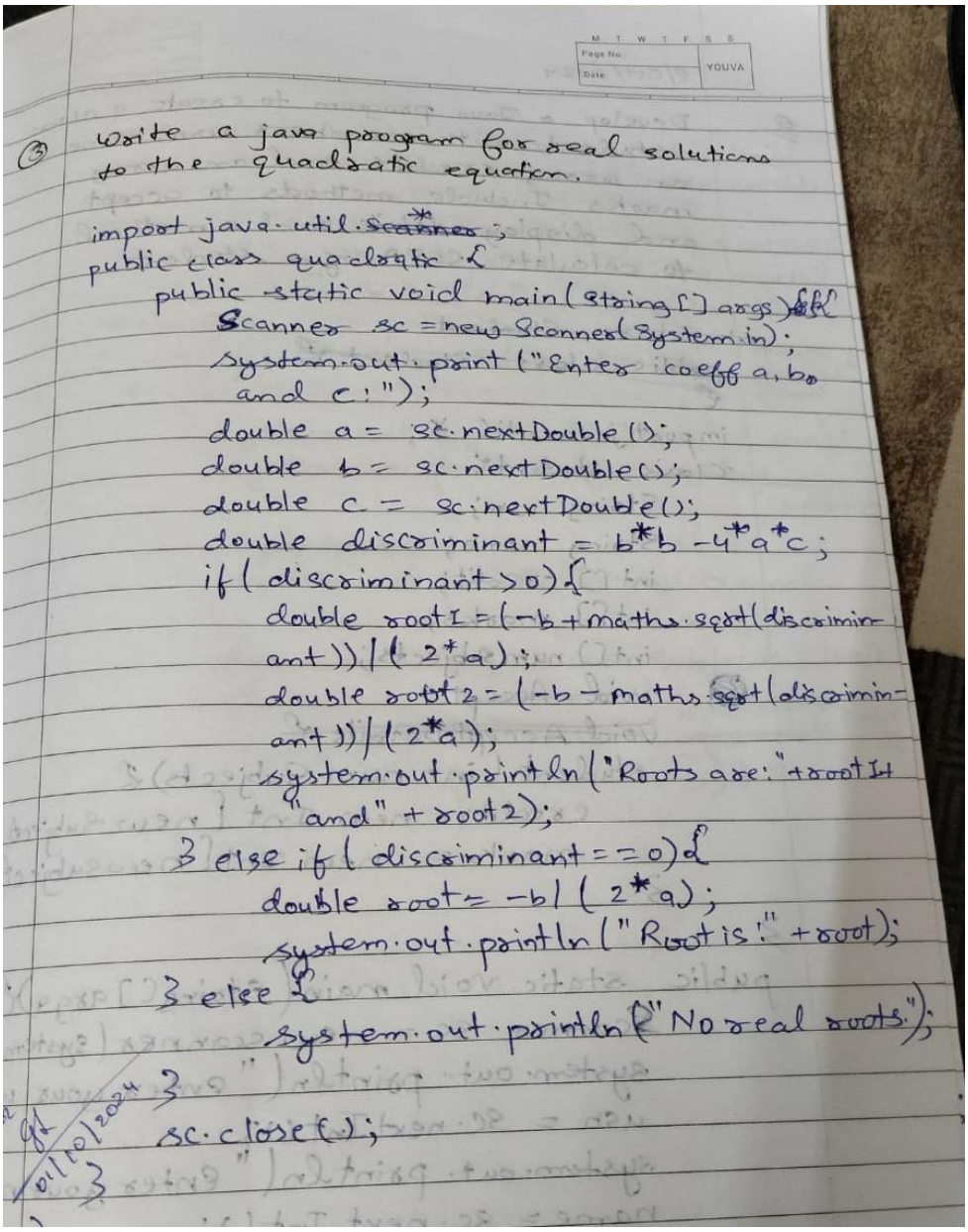
		each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.	
6	29/10/2024	Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the 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 d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.	
7	12/11/2024	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.	
8	19/11/2024	We have created an interface named Polygon. It includes a default method getPerimeter() and an abstract method getArea(). We can calculate the perimeter of all polygons in the same manner so we implemented the body of getPerimeter() in Polygon. Now, all polygons that implement Polygon can use getPerimeter() to calculate perimeter. However, the rule for calculating the area is different for different polygons. Hence, getArea() is included without implementation. Any class that implements Polygon must provide an implementation ofgetArea()	

9	26/11/2024	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 Wrong Age( ) when the input age=father’s age.	
10	3/12/2024	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.	

Github Link:
https://github.com/1bm23cs063/Java-Lab-Program.git

Program 1
Implement Quadratic Equation

Algorithm:



③ Write a java program for real solutions to the quadratic equation.

```
import java.util.*;
public class quadratic {
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter coeff a, b and c:");
        double a = sc.nextDouble();
        double b = sc.nextDouble();
        double c = sc.nextDouble();
        double discriminant = b*b -4*a*c;
        if(discriminant>0){
            double root1 = (-b+maths.sqrt(discriminant))/(2*a);
            double root2 = (-b-maths.sqrt(discriminant))/(2*a);
            system.out.println("Roots are:"+root1+"and"+root2);
        }else if(discriminant==0){
            double root= -b/(2*a);
            system.out.println("Root is:"+root);
        }else{
            system.out.println("No real roots.");
        }
        sc.close();
    }
}
```

01/10/2024

Code:

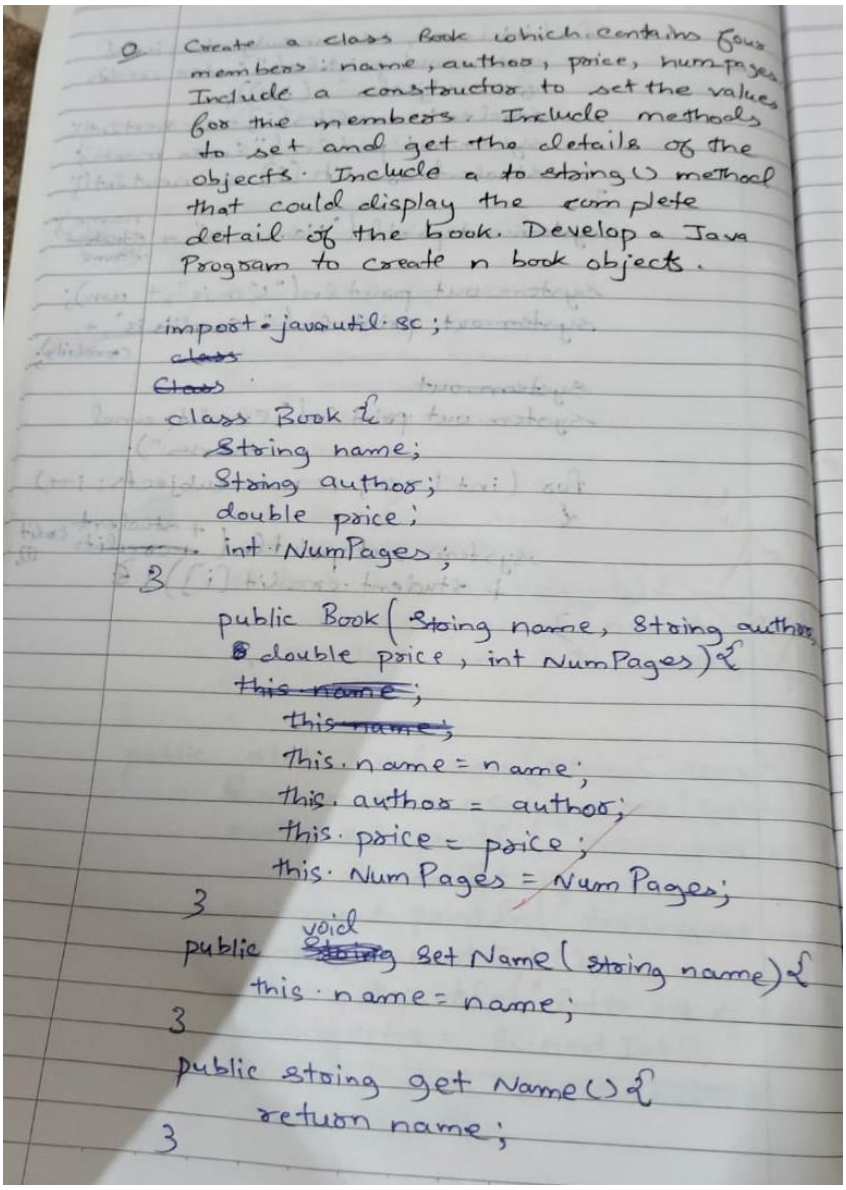
```
import java.util.Scanner;

public class Quadratic {
    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 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("Roots are: " + root1 + " and " + root2);
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("Root is: " + root);
        } else {
            System.out.println("No real roots.");
        }

        sc.close();
    }
}
```

Output:

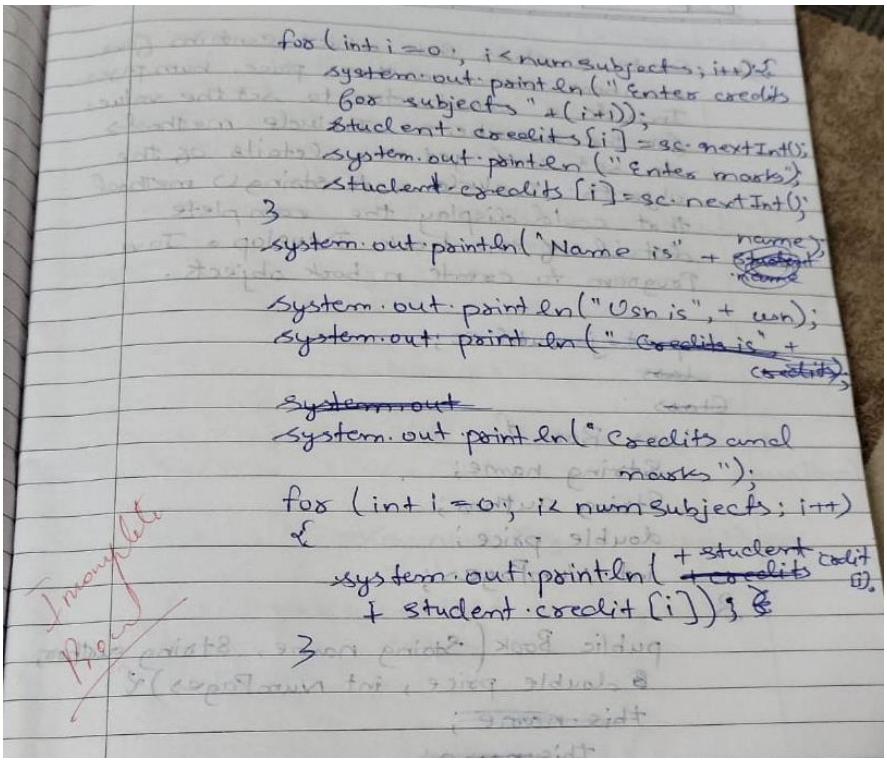
```
PS C:\Users\Acer\Desktop\bm23\java\New folder> javac Quadratic.java
PS C:\Users\Acer\Desktop\bm23\java\New folder> java Quadratic.java
Enter coefficient a: 2
Enter coefficient b: 1
Enter coefficient c: 1
No real roots.
```

Program 2
SGPA of student
Algorithm:



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

```
impost javautil.sc;
class
Class
class Book {
    String name;
    String author;
    double price;
    int NumPages;
}
    public Book( String name, String author,
    double price, int NumPages){
    this.name;
        this.name;
        This.name = name;
        this.author = author;
        this.price = price;
        this.NumPages = NumPages;
}
public void setName( string name){
    this.name = name;
}
public string getName(){
    return name;
}
```



Code:

```
import java.util.Scanner;

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

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

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

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

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

```
    System.out.println("Enter details for " + numSubjects + " subjects:");
    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();
    }
}

public void displayDetails() {
    System.out.println("\nStudent Details:");
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    System.out.println("Subjects Details:");
    for (int i = 0; i < numSubjects; i++) {
        System.out.println("Subject " + (i + 1) + ": Credits = " + credits[i] + ", Marks = " + marks[i]);
    }
}

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

    for (int i = 0; i < numSubjects; i++) {
        int gradePoint = getGradePoint(marks[i]);
        totalCredits += credits[i];
        totalPoints += gradePoint * credits[i];
    }

    return (double) totalPoints / totalCredits;
}

private int getGradePoint(int marks) {
    if (marks >= 90) return 10;
    else if (marks >= 80) return 9;
    else if (marks >= 70) return 8;
    else if (marks >= 60) return 7;
    else if (marks >= 50) return 6;
    else if (marks >= 40) return 5;
    else return 0; // Fail grade
}

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

    System.out.print("Enter the number of subjects: ");
```

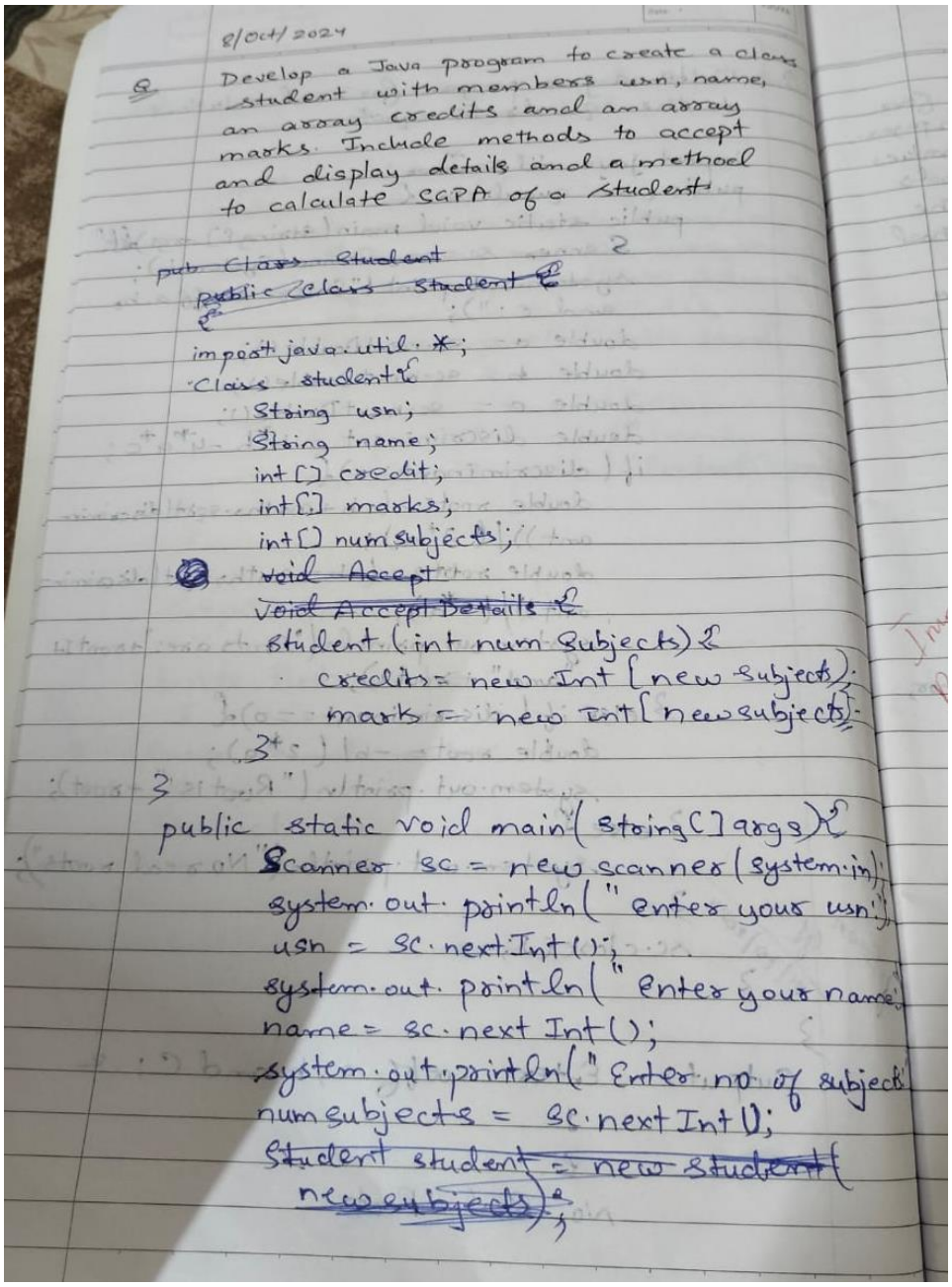
```
        int numSubjects = sc.nextInt();

        Student student = new Student(numSubjects);

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

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

Output:

```
PS C:\Users\Acer\Desktop\bm23\java\New folder> javac Student.java
PS C:\Users\Acer\Desktop\bm23\java\New folder> java Student.java
Enter the number of subjects: 3
Enter your USN: 123
Enter your Name: hgh
Enter details for 3 subjects:
Enter credits for subject 1: 4
Enter marks for subject 1: 89
Enter credits for subject 2: 4
Enter marks for subject 2: 89
Enter credits for subject 3: 4
Enter marks for subject 3: 89

Student Details:
USN: 123
Name: hgh
Subjects Details:
Subject 1: Credits = 4, Marks = 89
Subject 2: Credits = 4, Marks = 89
Subject 3: Credits = 4, Marks = 89
SGPA: 9.00
```

Program 3
Book Information

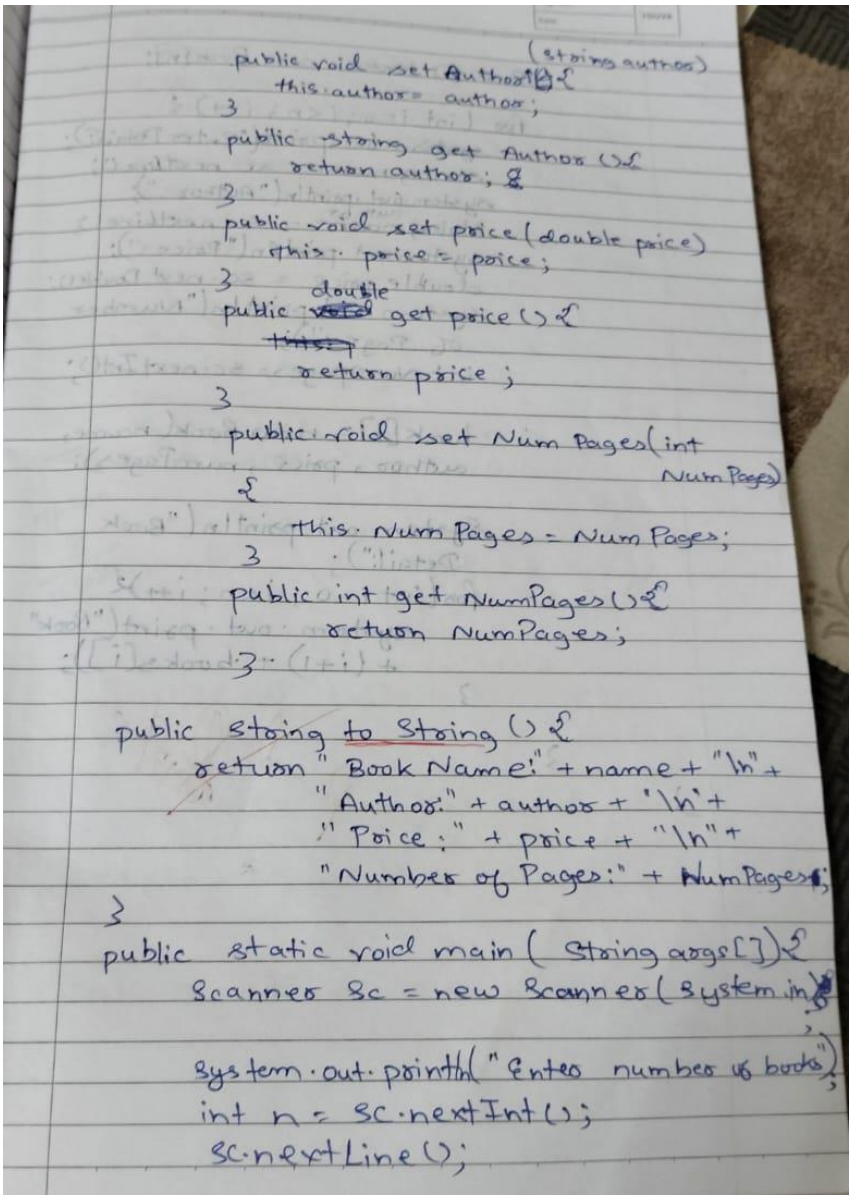
Algorithm:



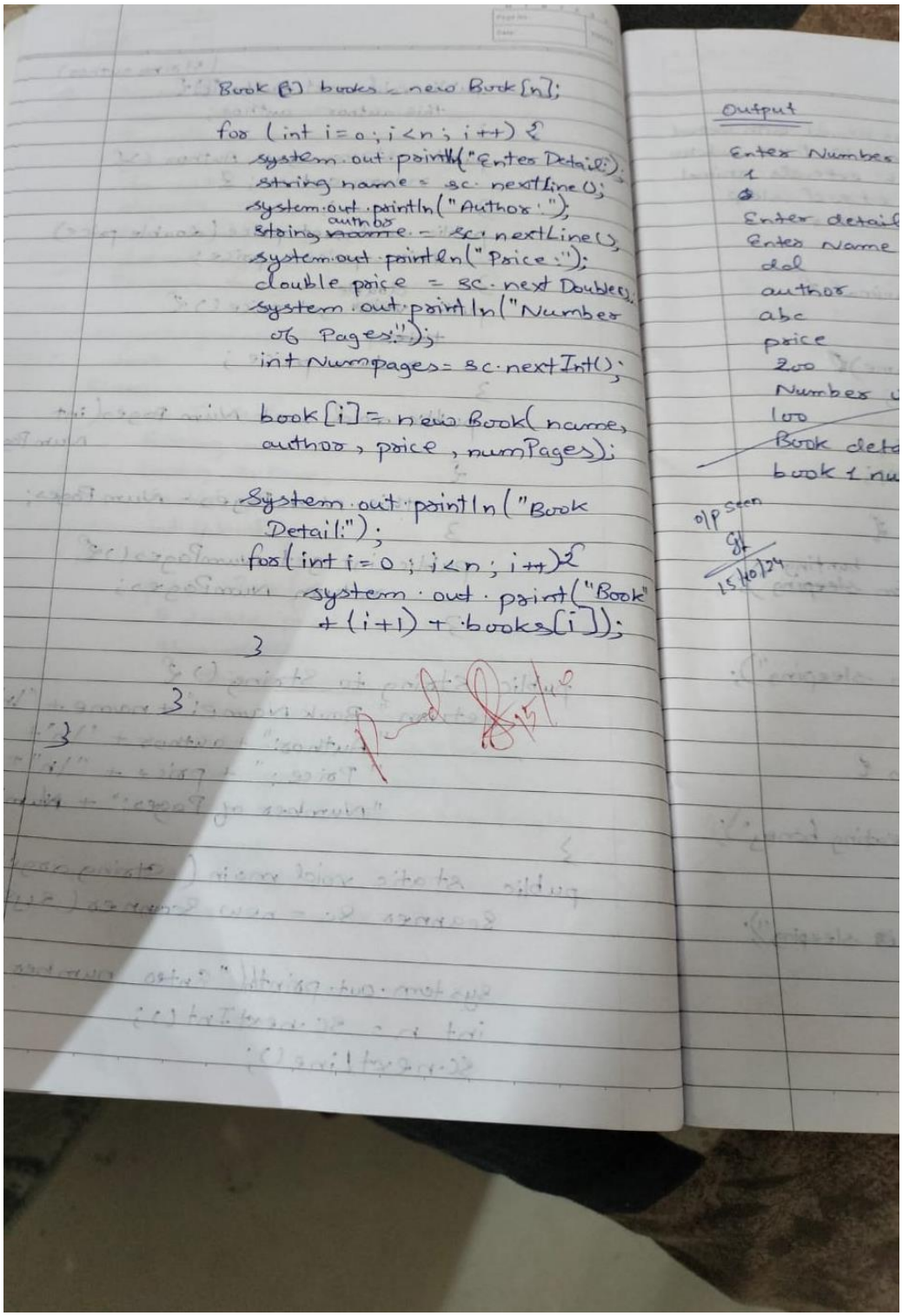
8/Oct/2024

Q 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.*;
class student{
    String usn;
    String name;
    int[] credit;
    int[] marks;
    int[] numsubjects;
    student(int numsubjects){
        credits = new Int[newsubjects];
        marks = new Int[newsubjects];
    }
}
public static void main(String[] args){
    Scanner sc = new scanner(system.in);
    system.out.println("enter your usn");
    usn = sc.nextInt();
    system.out.println("enter your name");
    name = sc.nextInt();
    system.out.println("Enter no of subjects");
    numsubjects = sc.nextInt();
```



```
public void setAuthor(String author){
    this.author = author;
}
public String getAuthor(){
    return author;
}
public void setPrice(double price)
    this.price = price;
}
public double getPrice(){
    return price;
}
public void setNumPages(int NumPages)
{
    this.NumPages = NumPages;
}
public int getNumPages(){
    return NumPages;
}

public String toString(){
    return "Book Name:" + name + "\n" +
           "Author:" + author + "\n" +
           "Price:" + price + "\n" +
           "Number of Pages:" + NumPages;
}
public static void main(String args[]){
    Scanner sc = new Scanner(System.in);

    System.out.println("Enter number of books");
    int n = sc.nextInt();
    sc.nextLine();
```



Code:

```
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 void setName(String name) {
   this.name = name;
}

public String getAuthor() {
   return author;
}

public void setAuthor(String author) {
   this.author = author;
}

public double getPrice() {
   return price;
}

public void setPrice(double price) {
   this.price = price;
}

public int getNumPages() {
   return numPages;
}

public void setNumPages(int numPages) {
   this.numPages = numPages;
}

public String toString() {
   return "Book Name: " + name + "\n" +
         "Author: " + author + "\n" +
```

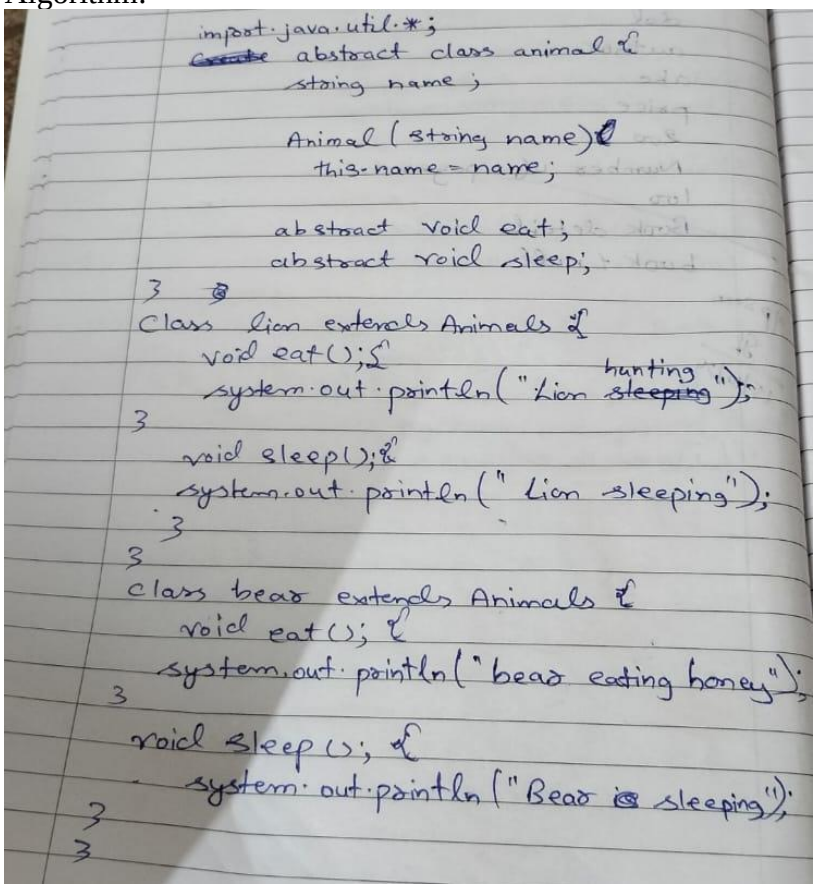
```
            "Price: " + price + "\n" +
            "Number of Pages: " + numPages + "\n";
    }
}

public class BookMain {
    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();

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for Book " + (i + 1) + ":");

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

            System.out.print("Enter Author: ");
            String author = sc.nextLine();

            System.out.print("Enter Price: ");
            double price = sc.nextDouble();

            System.out.print("Enter Number of Pages: ");
            int numPages = sc.nextInt();
            sc.nextLine(); // Consume the newline character

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

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

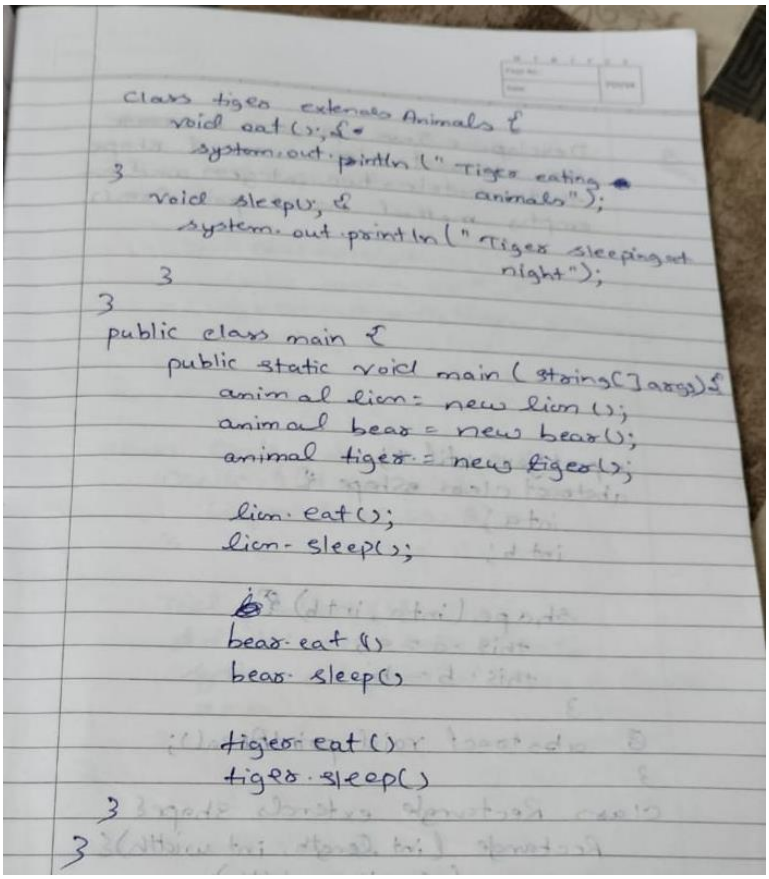
Output:

```
PS C:\Users\Acer\Desktop\bm23\java\New folder> javac BookMain.java
PS C:\Users\Acer\Desktop\bm23\java\New folder> java BookMain.java
error: can't find main(String[]) method in class: Book
PS C:\Users\Acer\Desktop\bm23\java\New folder> java BookMain
Enter the number of books: 1
Enter details for Book 1:
Enter Name: dd
Enter Author: abc
Enter Price: 200
Enter Number of Pages: 100

Book Details:
Book 1:
Book Name: dd
Author: abc
Price: 200.0
Number of Pages: 100
```

Program 4
Abstract Class Animal

Algorithm:





Code: import java.util.*;

```
abstract class Animal {
   String name;

   Animal(String name) {
      this.name = name;
   }

   abstract void eat();

   abstract void sleep();
}

class Lion extends Animal {
   Lion(String name) {
      super(name);
   }

   void eat() {
      System.out.println(name + " is eating meat.");
   }
```

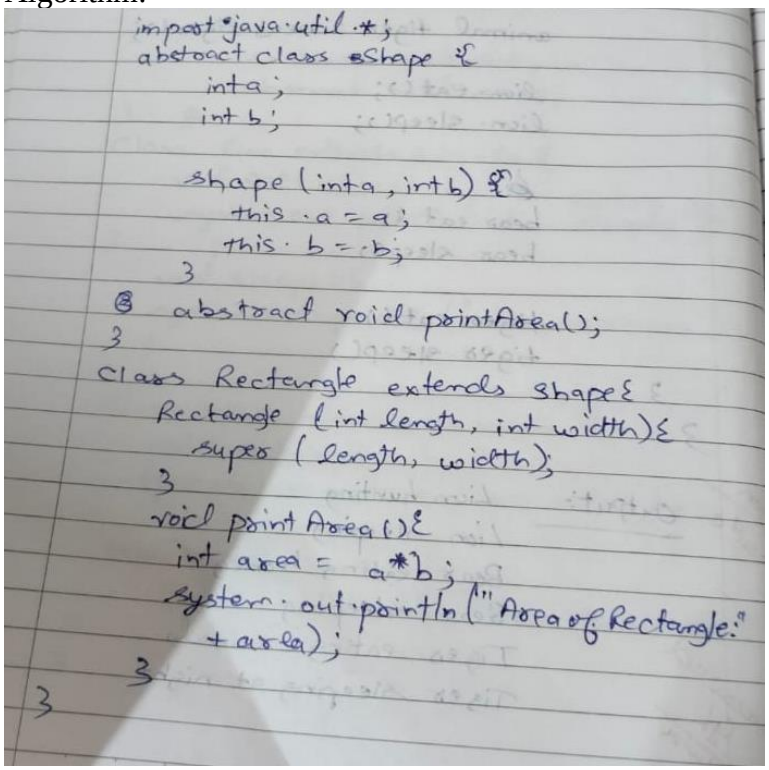
```
    void sleep() {
        System.out.println(name + " is sleeping.");
    }
}

class Bear extends Animal {
    Bear(String name) {
        super(name);
    }

    void eat() {
        System.out.println(name + " is eating honey.");
    }

    void sleep() {
        System.out.println(name + " is sleeping.");
    }
}

class Tiger extends Animal {
    Tiger(String name) {
        super(name);
    }

    void eat() {
        System.out.println(name + " is eating other animals.");
    }

    void sleep() {
        System.out.println(name + " is sleeping at night.");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal lion = new Lion("Lion");
        Animal bear = new Bear("Bear");
        Animal tiger = new Tiger("Tiger");

        lion.eat();
        lion.sleep();

        bear.eat();
        bear.sleep();

        tiger.eat();
        tiger.sleep();
```

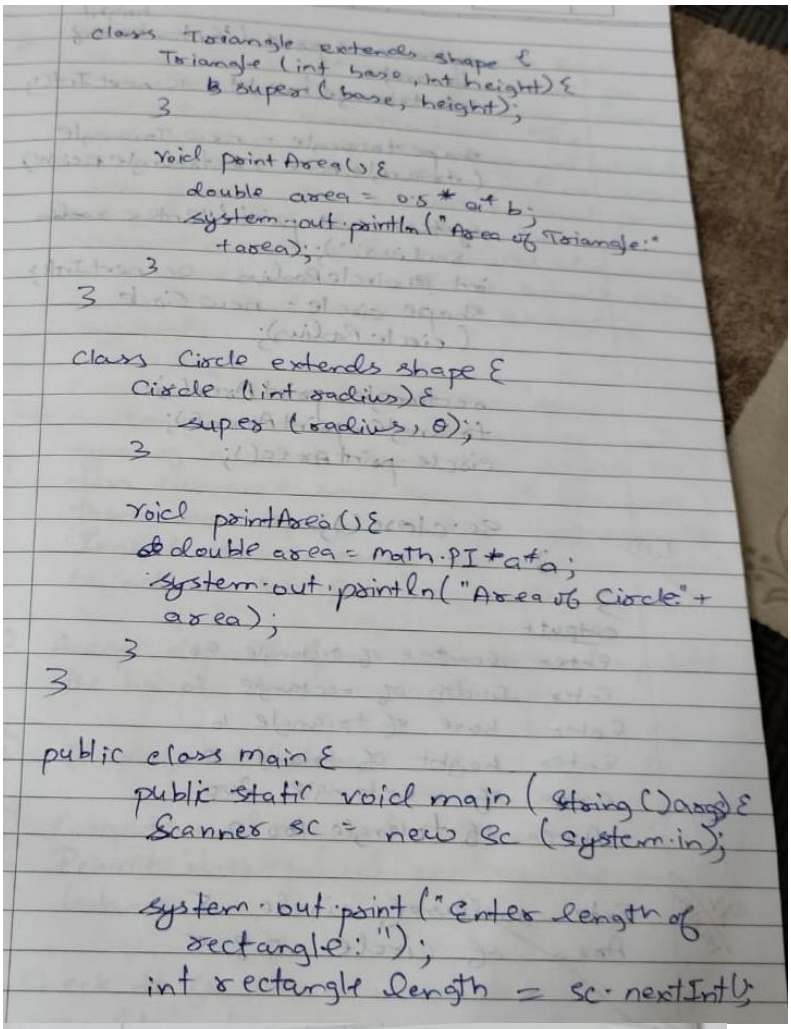
```
    }
}
```

Output:

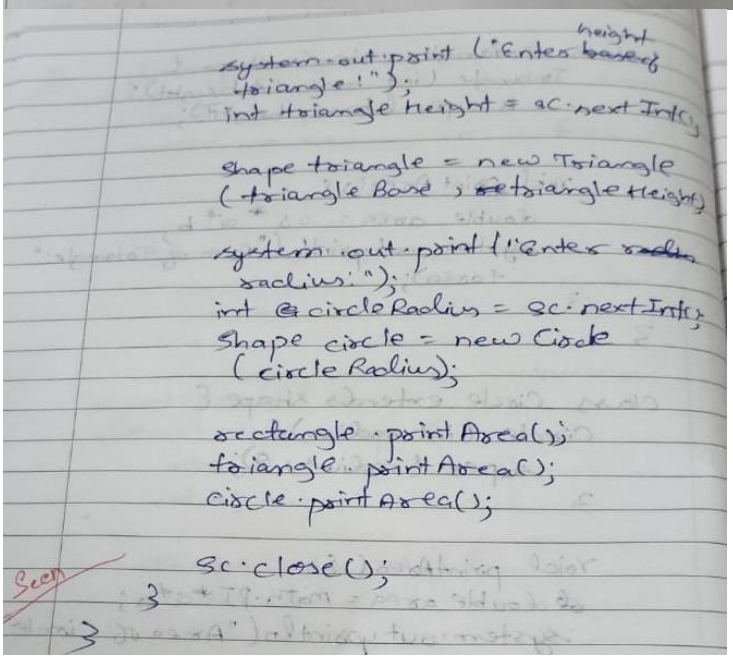
```
PS C:\Users\Acer\Desktop\bm23\java\New folder> javac Main.java
PS C:\Users\Acer\Desktop\bm23\java\New folder> java Main.java
error: can't find main(String[]) method in class: Animal
PS C:\Users\Acer\Desktop\bm23\java\New folder> java Main
Lion is eating meat.
Lion is sleeping.
Bear is eating honey.
Bear is sleeping.
Tiger is eating other animals.
Tiger is sleeping at night.
```

Program 5:
Abstract Class Shape
Algorithm:





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

    void printArea() {
        double area = 0.5 * a * b;
        System.out.println("Area of Triangle:"
            + area);
    }
}

class Circle extends Shape {
    Circle (int radius) {
        super (radius, 0);
    }

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

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

        System.out.print ("Enter length of
            rectangle: ");
        int rectangle length = sc.nextInt();
```



```
        System.out.print ("Enter height of
            triangle:");
        int triangle Height = sc.nextInt();

        Shape triangle = new Triangle
        (triangle Base, triangle Height);

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

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

        sc.close();
    }
}
```

Seen

Code:

```
import java.util.*;

abstract class Shape {
   int a;
   int b;

   Shape(int a, int b) {
      this.a = a;
      this.b = b;
   }

   abstract void printArea();
}

class Rectangle extends Shape {
   Rectangle(int length, int width) {
      super(length, width);
   }

   void printArea() {
      int area = a * b;
      System.out.println("Area of Rectangle: " + area);
   }
}

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

   void printArea() {
      double area = 0.5 * a * b;
      System.out.println("Area of Triangle: " + area);
   }
}

class Circle extends Shape {
   Circle(int radius) {
      super(radius, 0); // `b` is not used for Circle
   }

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

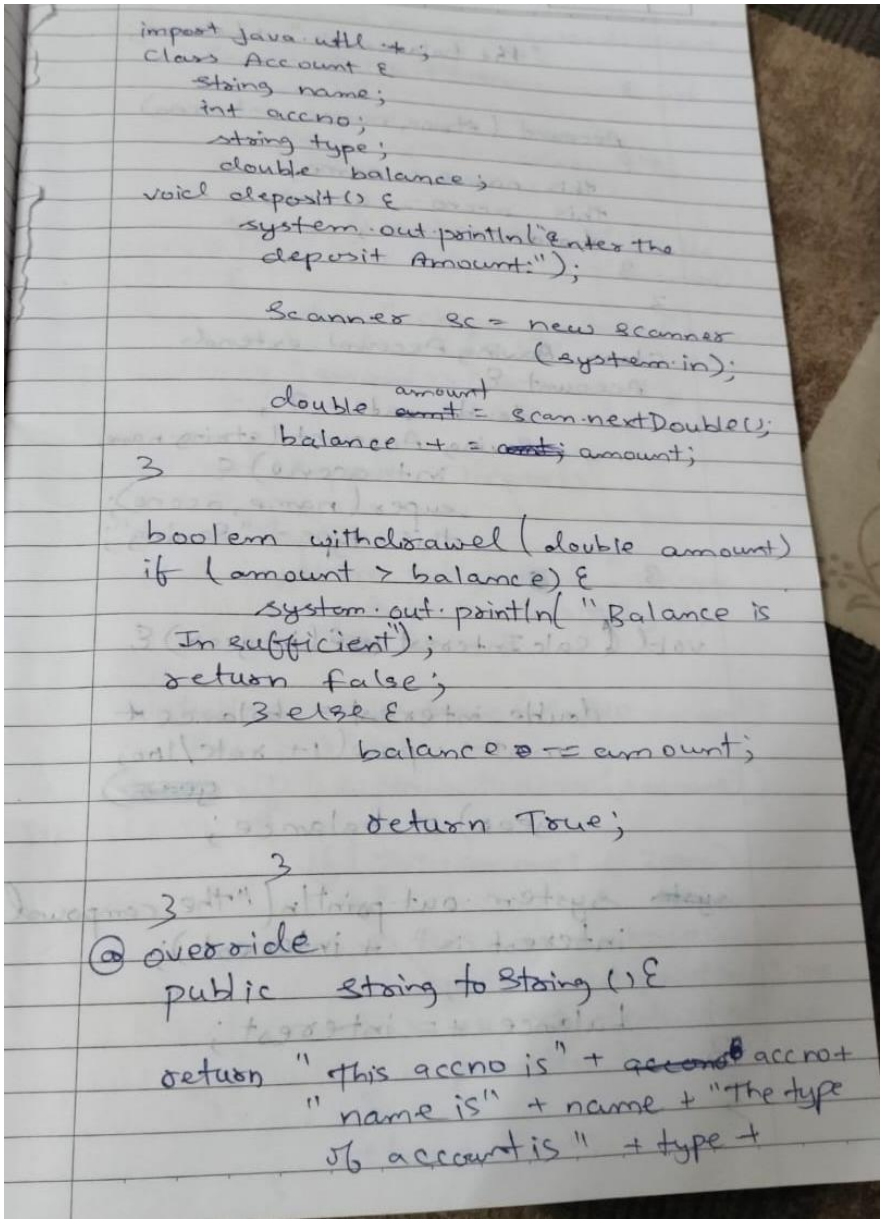
```
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

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

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

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

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

        sc.close();
    }
}
```

Output:

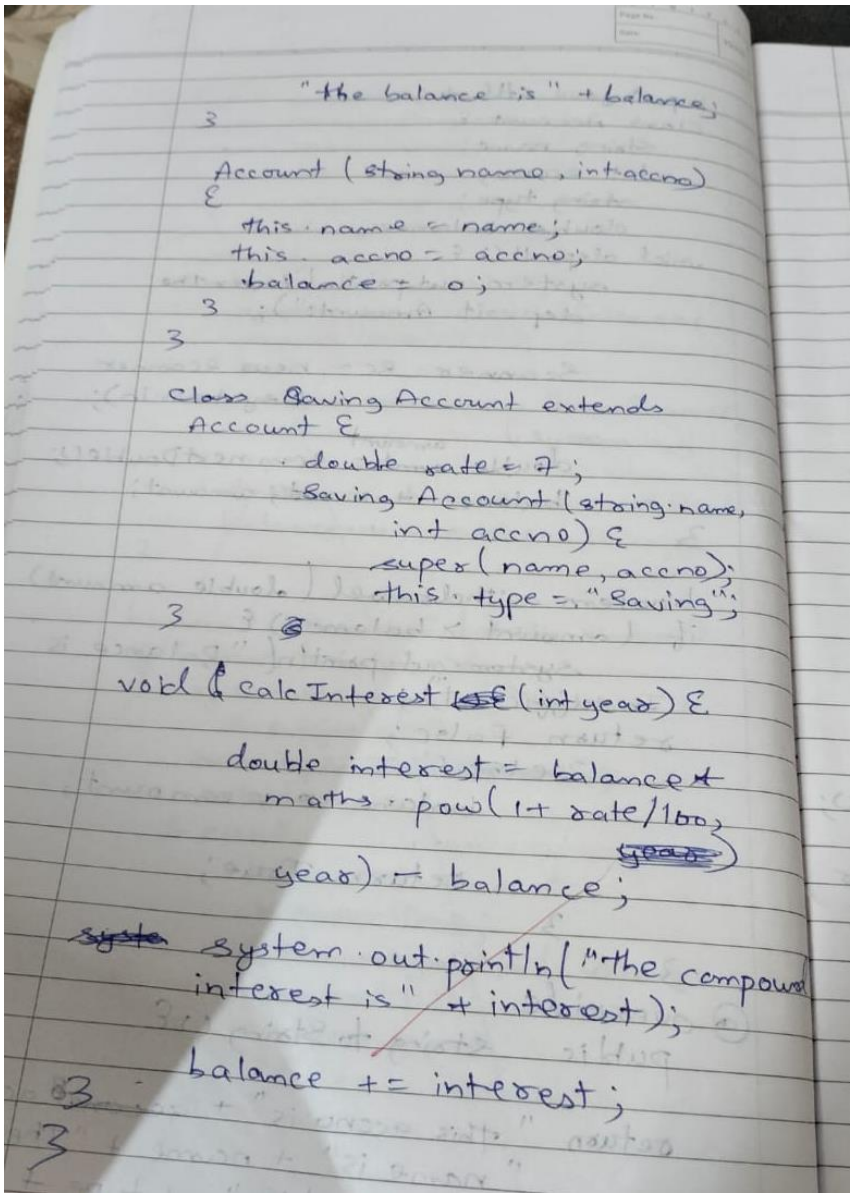
```
Enter length of rectangle: 20
Enter width of rectangle: 10
Enter base of triangle: 10
Enter height of triangle: 20
Enter radius of circle: 5
Area of Rectangle: 200
Area of Triangle: 100.0
Area of Circle: 78.53981633974483
```

Program 6

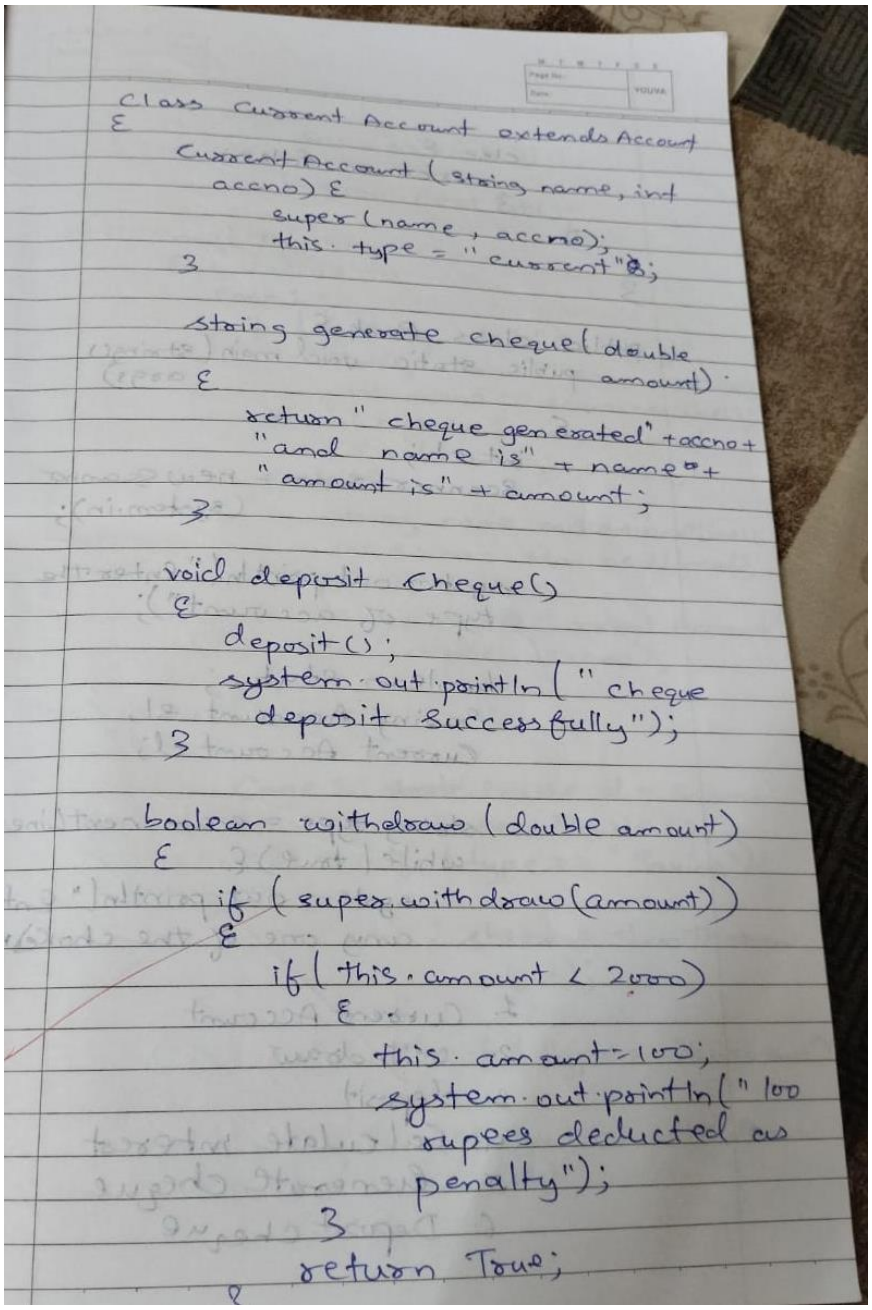
Bank Class
Algorithm:



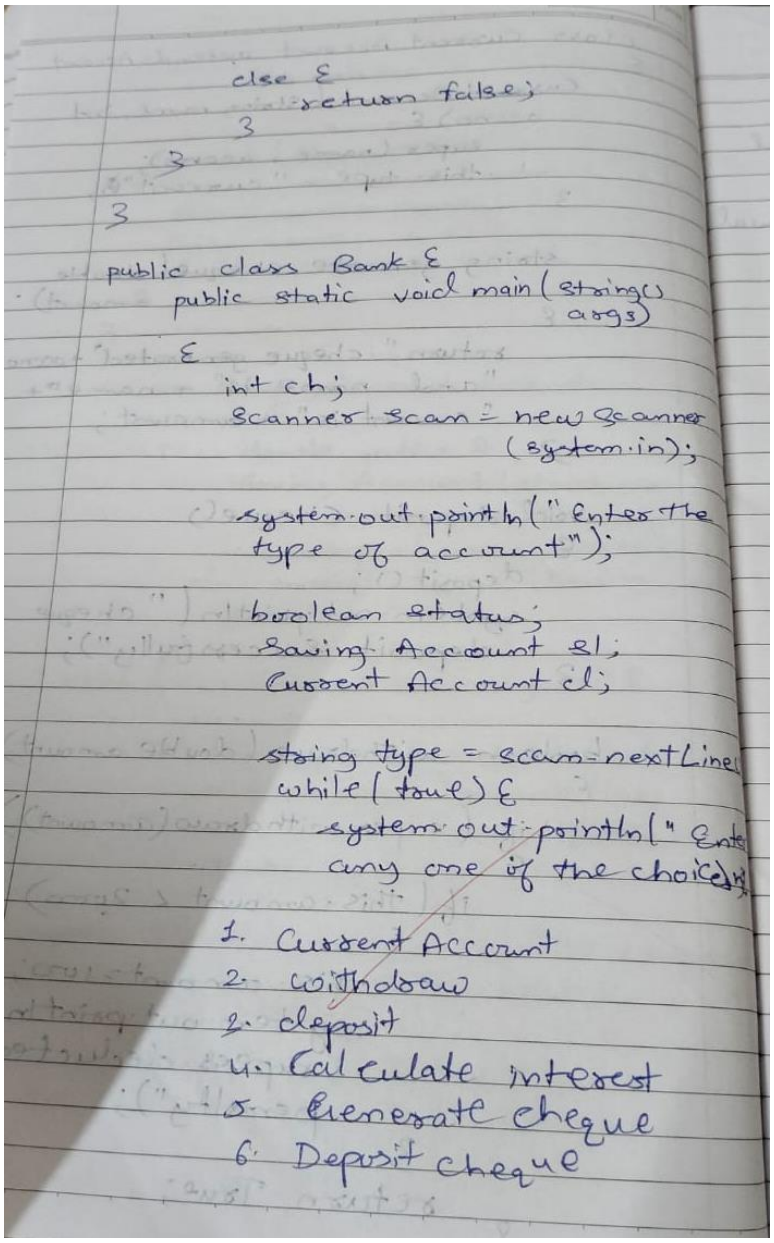
```
import java.util.*;
Class Account {
    String name;
    int accno;
    string type;
    double balance;
voicl deposit() {
    system.out.println("Enter the
    deposit Amount:");

    Scanner sc = new scanner
                 (system.in);
    double amount
    ~~amt~~ = scan.nextDouble();
    balance += ~~amt~~ amount;
}

boolem withdrawel (double amount)
if (amount > balance) {
    System.out.println("Balance is
In sufficient");
return false;
} else {
    balance -= amount;

    return True;
}
}
@override
public String to String () {

return "this accno is" + accno+
    " name is" + name + "The type
    of account is" + type +
```



```
        "the balance is" + balance);
    }

    Account (string name, int accno)
    {
        this.name = name;
        this.accno = accno;
        balance = 0;
    }
}

class Saving Account extends
    Account {
        double rate = 7;
        Saving Account (string name,
                int accno) {
            super(name, accno);
            this.type = "saving";
    }
void calc Interest (int year) {

    double interest = balance *
        maths.pow(1 + rate/100,
                year) - balance;

    system.out.println("the compound
        interest is" + interest);

    balance += interest;
}
}
```

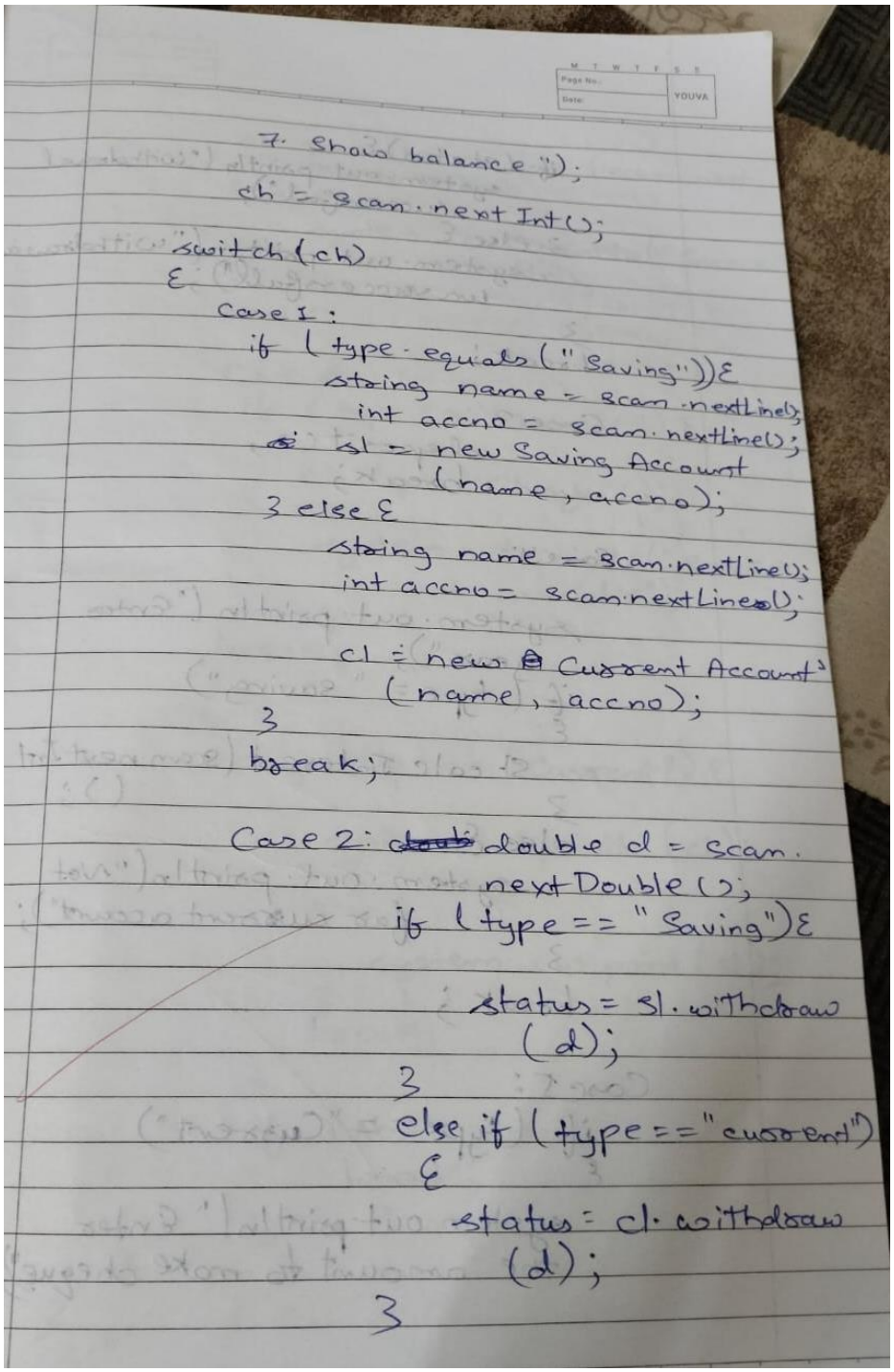


```
class Current Account extends Account
{
    Current Account (string name, int
    accno) {
        super (name, accno);
        this. type = "current";
    }

    string generate cheque (double
                            amount).
    {
        return " cheque generated" + accno +
        "and name is" + name +
        "amount is" + amount;
    }

    void deposit Cheque()
    {
        deposit();
        system. out. println ( " cheque
        deposit successfully");
    }

    boolean withdraw (double amount)
    {
        if (super. withdraw (amount))
        {
            if ( this. amount < 2000)
            {
                this. amount = 100;
                system. out. println ( " 100
                rupees deducted as
                penalty");
            }
            return True;
```

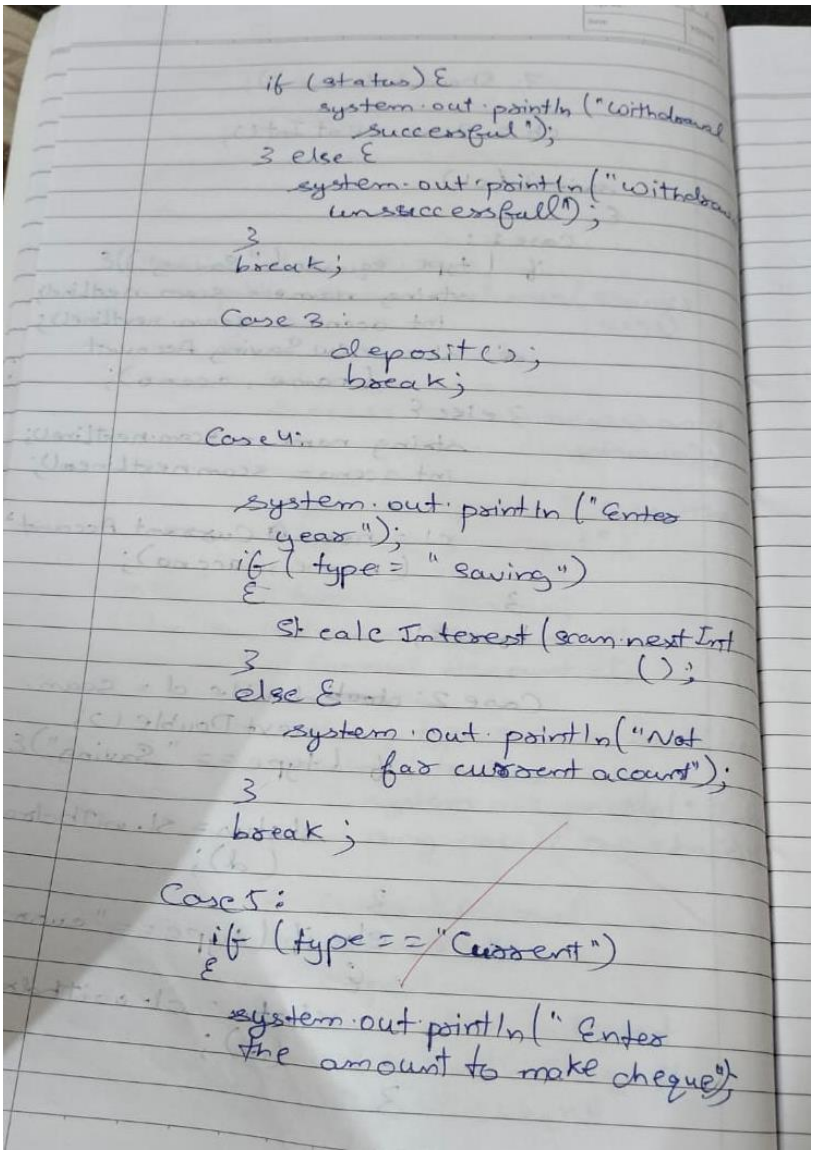


```
        else {
            return false;
        }
    }
}

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

        System.out.println("Enter the
        type of account");

        boolean status;
        Saving Account sl;
        Current Account cl;

        String type = scan.nextLine
        while(true) {
            System.out.println("Ente
            any one of the choices");
```

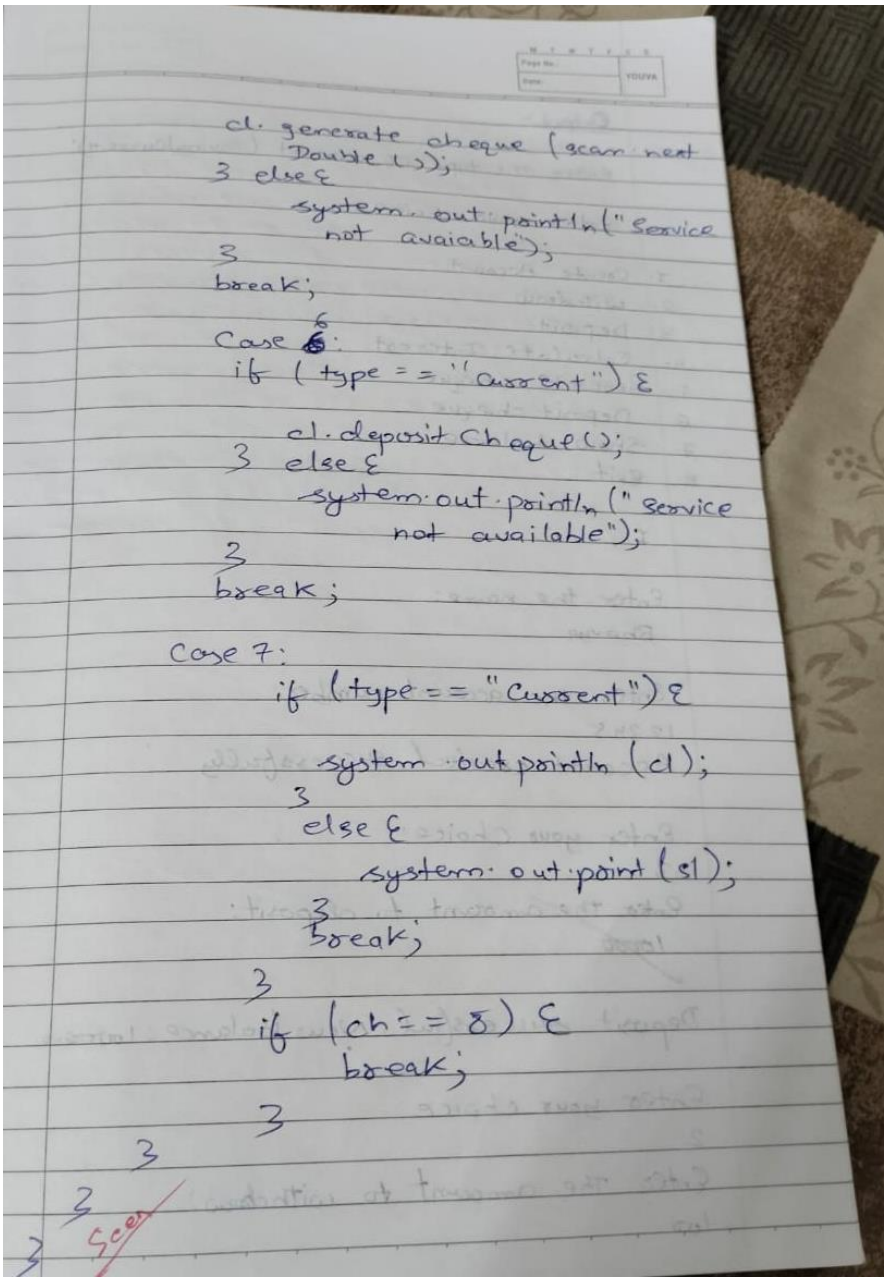
1. Current Account
2. withdraw
3. deposit
4. Calculate interest
5. Generate cheque
6. Deposit cheque



```
7. Show balance");
ch = scan.nextInt();
switch(ch)
{
  case 1:
    if (type.equals("Saving")){
      string name = scan.nextLine();
      int accno = scan.nextLine();
      sl = new Saving Account
           (name, accno);
    } else {
      string name = scan.nextLine();
      int accno = scan.nextLine();
      cl = new Current Account
           (name, accno);
    }
  break;

  Case 2: double d = scan.
          nextDouble();
          if (type == "Saving"){

            status = sl.withdraw
                     (d);
          }
          else if (type == "current")
          {
            status = cl.withdraw
                     (d);
          }
```



```
if (status) {
    system.out.println("withdrawal
        successful");
} else {
    system.out.println("withdraw
        unsuccessfull");
}
break;

Case 3:
    deposit();
    break;

Case 4:

    System.out.println("Enter
        year");
    if (type = "saving")
    {
        St calcInterest(scan.nextInt
                        ());
    }
    else {
        system.out.println("Not
            for current account");
    }
    break;

Case 5:
    if (type == "Current")
    {
        system.out.println("Enter
        the amount to make cheque");
```



Code:

```
mport java.util.*;

public class Main {
   public static void main(String[] args){
      int ch;
      String type;
      Scanner scan = new Scanner(System.in);
      System.out.println("Enter the type of account (Saving/Current):");
      type = scan.nextLine();
      boolean status;
      SavingAccount s1 = new SavingAccount();
      CurrentAccount c1 = new CurrentAccount();
```

```
        while (true) {
            System.out.println("Enter your choice: \n 1. Create Account \n 2. Withdraw \n 3. Deposit \n 4. Calculate Interest \n 5. Generate Cheque \n 6. Deposit Cheque \n 7. Show Balance \n 8. Exit");
            ch = scan.nextInt();
            switch (ch) {
                case 1:
                    System.out.println("Enter the name:");
                    String name = scan.next();
                    System.out.println("Enter the account number:");
                    int accno = scan.nextInt();
                    if(type.equalsIgnoreCase("saving")){
                        s1 = new SavingAccount(name, accno);
                    } else {
                        c1 = new CurrentAccount(name, accno);
                    }
                    System.out.println("Account created successfully");
                    break;

                case 2:
                    System.out.println("Enter the amount to withdraw:");
                    double amount = scan.nextDouble();
                    if(type.equalsIgnoreCase("saving")) {
                        status = s1.withdraw(amount);
                    } else {
                        status = c1.withdraw(amount);
                    }
                    System.out.println(status ? "Withdraw completed successfully" : "Withdraw unsuccessful");
                    break;

                case 3:
                    if(type.equalsIgnoreCase("saving")) {
                        s1.deposit();
                    } else {
                        c1.deposit();
                    }
                    break;

                case 4:
                    if(type.equalsIgnoreCase("saving")) {
                        s1.calcInterest();
                    } else {
                        System.out.println("Service not available");
                    }
                    break;
```

```
            case 5:
                if(type.equalsIgnoreCase("current")) {
                    System.out.println("Enter the amount to generate cheque:");
                    System.out.println(c1.generateCheque(scan.nextDouble()));
                } else {
                    System.out.println("Service not available");
                }
                break;

            case 6:
                if(type.equalsIgnoreCase("current")) {
                    c1.depositCheque();
                } else {
                    System.out.println("Service not available");
                }
                break;

            case 7:
                if(type.equalsIgnoreCase("saving")) {
                    System.out.println(s1);
                } else {
                    System.out.println(c1);
                }
                break;

            case 8:
                System.out.println("Exiting the program.");
                scan.close();
                return;

            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
  }
}

class Account {
    String name;
    String type;
    int accno;
    double balance;

    Account() {}

    Account(String name, int accno) {
        this.name = name;
```

```
        this.accno = accno;
        this.balance = 0;
    }

    void deposit() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the amount to deposit:");
        double amount = scanner.nextDouble();
        balance += amount;
        System.out.println("Deposit successful. New balance: " + balance);
    }

    boolean withdraw(double amount) {
        if(balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
            return true;
        } else {
            System.out.println("Insufficient balance.");
            return false;
        }
    }

    public String toString() {
        return "Account [name=" + name + ", type=" + type + ", accno=" + accno + ", balance=" +
balance + "]";
    }
}

class CurrentAccount extends Account {
    CurrentAccount() {}

    CurrentAccount(String name, int accno) {
        super(name, accno);
        type = "Current";
    }

    String generateCheque(double amount) {
        return "Cheque generated for Account: " + accno + ", Name: " + name + ", Amount: " + amount;
    }

    void depositCheque() {
        super.deposit();
        System.out.println("Cheque deposited successfully");
    }

    boolean withdraw(double amount) {
```

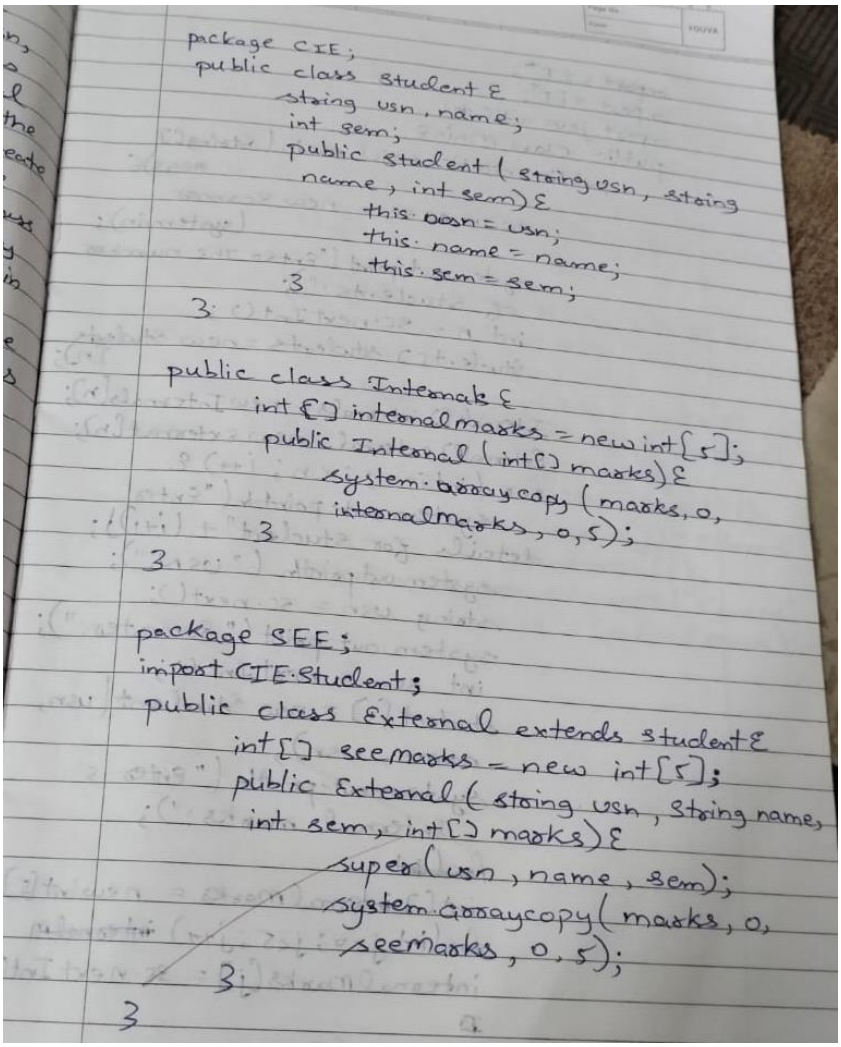
```
        if(super.withdraw(amount)) {
            if(balance < 2000) {
                System.out.println("100 rupees penalty for low balance");
            }
            return true;
        }
        return false;
    }
}

class SavingAccount extends Account {
    SavingAccount() {}

    SavingAccount(String name, int accno) {
        super(name, accno);
        type = "Saving";
    }

    void calcInterest() {
        double interest = balance * 0.05;
        System.out.println("Interest: " + interest + " has been deposited in your account");
        balance += interest;
    }
}
```

Output:

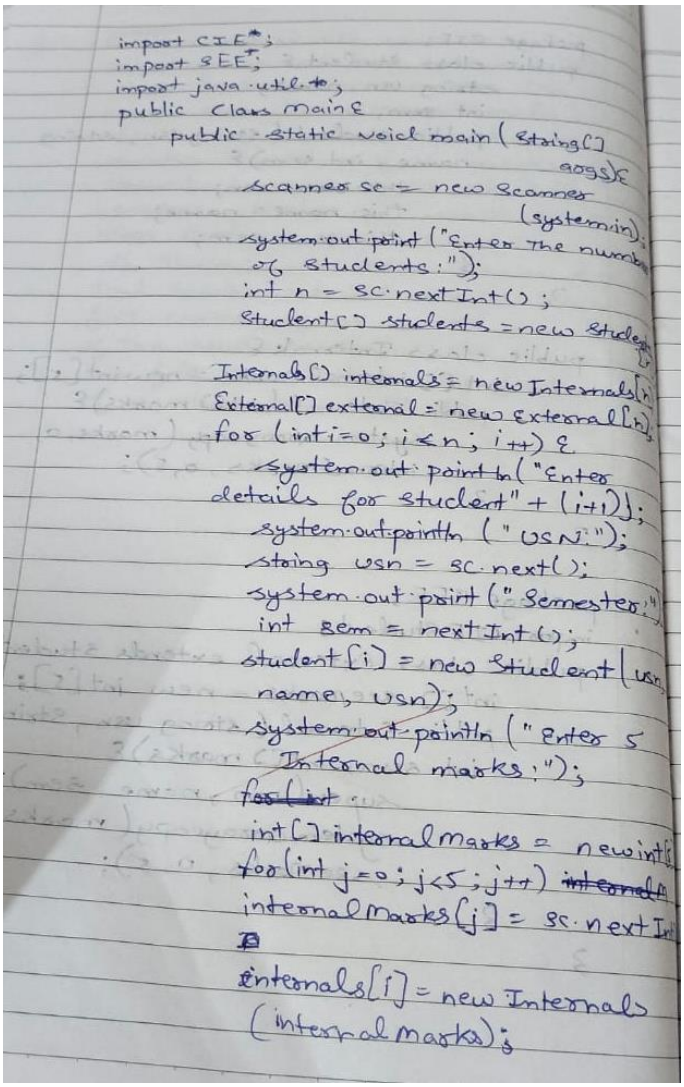
```
Enter the type of account (Saving/Current):
saving
Enter your choice:
 1. Create Account
 2. Withdraw
 3. Deposit
 4. Calculate Interest
 5. Generate Cheque
 6. Deposit Cheque
 7. Show Balance
 8. Exit
1
Enter the name:
bhavya
Enter the account number:
12345
Account created successfully
Enter your choice:
 1. Create Account
 2. Withdraw
 3. Deposit
 4. Calculate Interest
 5. Generate Cheque
 6. Deposit Cheque
 7. Show Balance
 8. Exit
3
Enter the amount to deposit:
10000
Deposit successful. New balance: 10000.0
```

```
Enter your choice:
 1. Create Account
 2. Withdraw
 3. Deposit
 4. Calculate Interest
 5. Generate Cheque
 6. Deposit Cheque
 7. Show Balance
 8. Exit
2
Enter the amount to withdraw:
100
Withdrawal successful. New balance: 9900.0
Withdraw completed successfully
Enter your choice:
 1. Create Account
 2. Withdraw
 3. Deposit
 4. Calculate Interest
 5. Generate Cheque
 6. Deposit Cheque
 7. Show Balance
 8. Exit
7
Account [name=bhavya, type=Saving, accno=12345, balance=9900.0]
Enter your choice:
 1. Create Account
 2. Withdraw
 3. Deposit
 4. Calculate Interest
 5. Generate Cheque
 6. Deposit Cheque
 7. Show Balance
 8. Exit
8
Exiting the program.
```

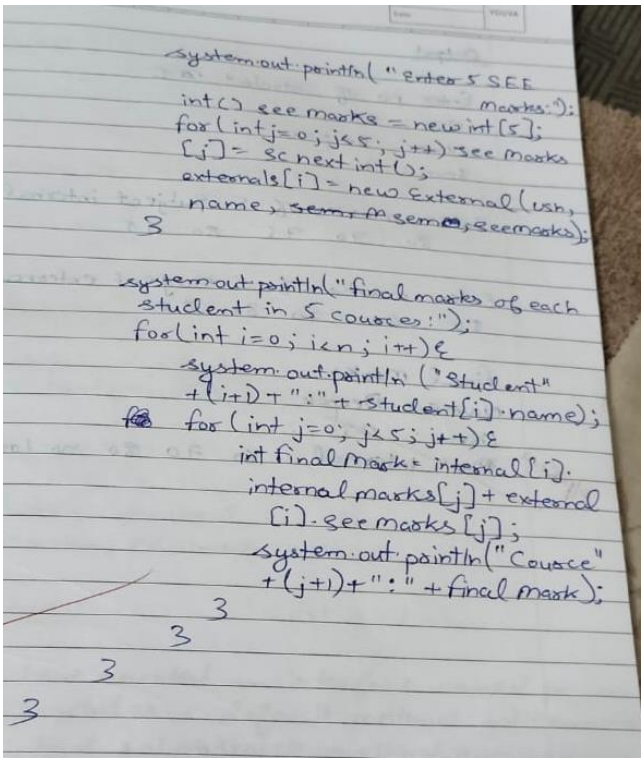
Program 7:
Package
Algorithm:



```
package CIE;
public class Student {
    string usn, name;
    int sem;
    public Student (string usn, string
    name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

public class Internals {
    int [] internalmarks = new int [5];
    public Internal (int[] marks) {
        System.arraycopy (marks, 0,
        internalmarks, 0, 5);
    }
}

package SEE;
import CIE.Student;
public class External extends student {
    int[] seemarks = new int[5];
    public External ( string usn, String name,
    int sem, int[] marks) {
        super (usn, name, sem);
        System.arraycopy( marks, 0,
        seemarks, 0, 5);
    }
}
```



```
import CIE*;
import SEE*;
import java.util.*;
public Class Main{
    public static voicl main(String[]
                                args){
        Scanner sc = new Scanner
                        (System.in);
        system.out.print("Enter the number
        of Students:");
        int n = sc.nextInt();
        Student[] students = new Student[
    Internals[] internals = new Internals[n];
    External[] external = new External[n];
    for(int i=0; i<n; i++){
        System.out.println("Enter
    details for student" + (i+1));
        system.out.println("USN:");
        staing usn = sc.next();
        system.out.print("Semester:");
        int sem = next Int();
        student[i] = new Student(usn,
        name, usn);
        System.out.println("Enter 5
            Internal marks:");
        for(int
        int[] internalMarks = new int[
        for(int j=0; j<5; j++) internal
        internalMarks[j] = sc.nextIn
    }
    internals[i] = new Internals
    (internalMarks);
```



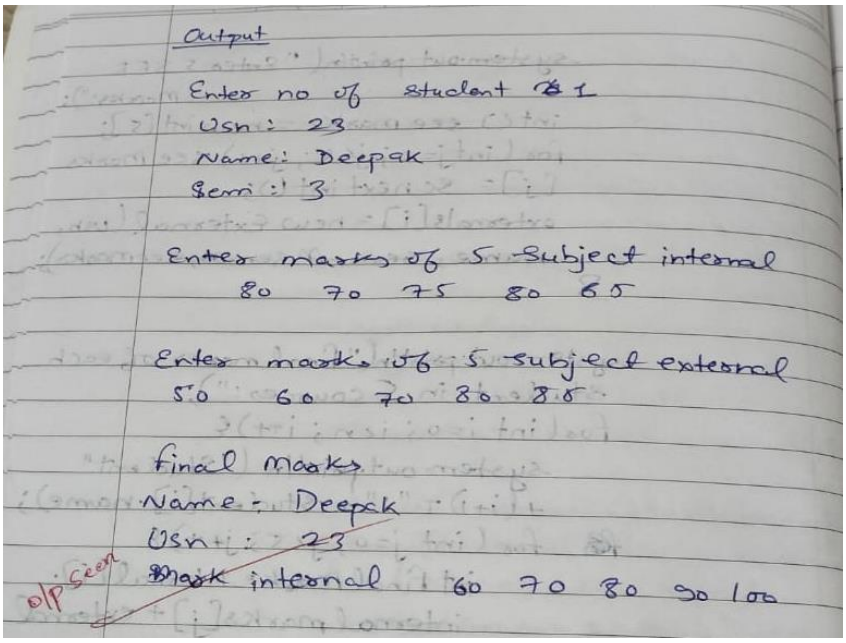
Code:

```
package CIE;
public class Student {
    String usn, name;
    int sem;
    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
public class Internals {
    int[] internalMarks = new int[5];
    public Internals(int[] marks) {
        System.arraycopy(marks, 0, internalMarks, 0, 5);
    }
}
```

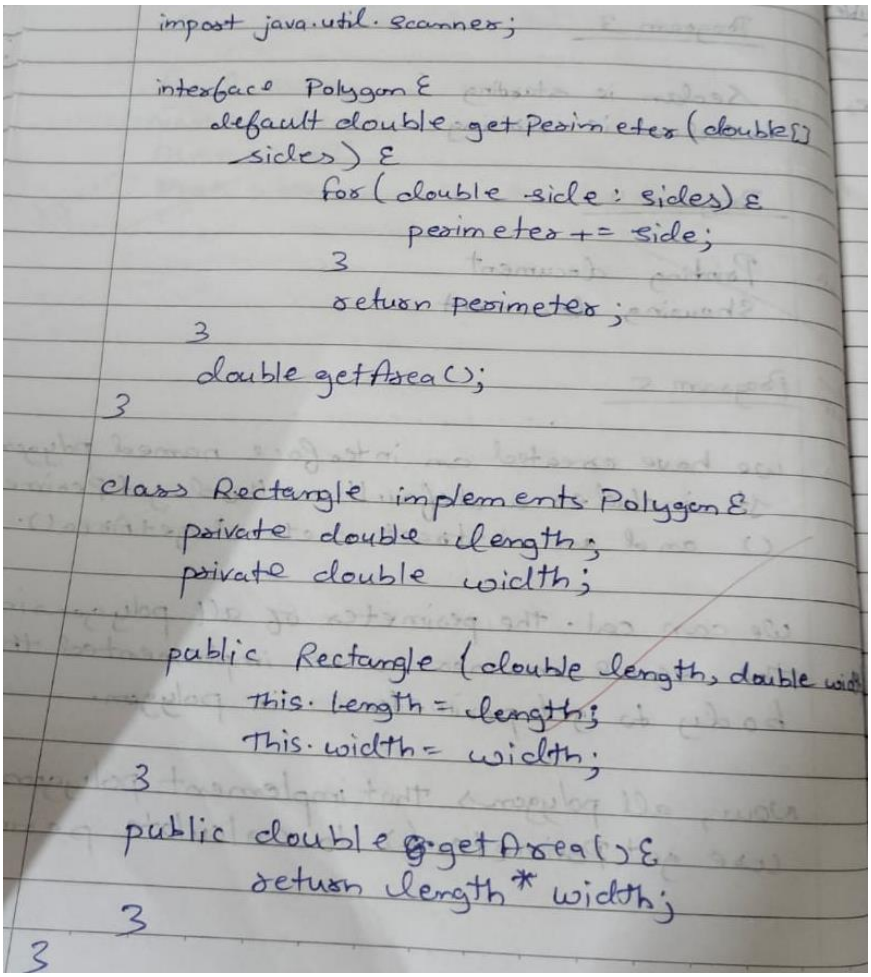
```
package SEE;
import CIE.Student;
public class External extends Student {
    int[] seeMarks = new int[5];
    public External(String usn, String name, int sem, int[] marks) {
        super(usn, name, sem);
        System.arraycopy(marks, 0, seeMarks, 0, 5);
    }
```

```
}

import CIE.*;
import SEE.*;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();
        Student[] students = new Student[n];
        Internals[] internals = new Internals[n];
        External[] externals = new External[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for student " + (i + 1));
            System.out.print("USN: ");
            String usn = sc.next();
            System.out.print("Name: ");
            String name = sc.next();
            System.out.print("Semester: ");
            int sem = sc.nextInt();
            students[i] = new Student(usn, name, sem);
            System.out.print("Enter 5 internal marks: ");
            int[] internalMarks = new int[5];
            for (int j = 0; j < 5; j++) internalMarks[j] = sc.nextInt();
            internals[i] = new Internals(internalMarks);
            System.out.print("Enter 5 SEE marks: ");
            int[] seeMarks = new int[5];
            for (int j = 0; j < 5; j++) seeMarks[j] = sc.nextInt();
            externals[i] = new External(usn, name, sem, seeMarks);
        }
        System.out.println("Final marks of each student in 5 courses:");
        for (int i = 0; i < n; i++) {
            System.out.println("Student " + (i + 1) + ": " + students[i].name);
            for (int j = 0; j < 5; j++) {
                int finalMark = internals[i].internalMarks[j] + externals[i].seeMarks[j];
                System.out.println("Course " + (j + 1) + ": " + finalMark);
            }
        }
    }
}
```

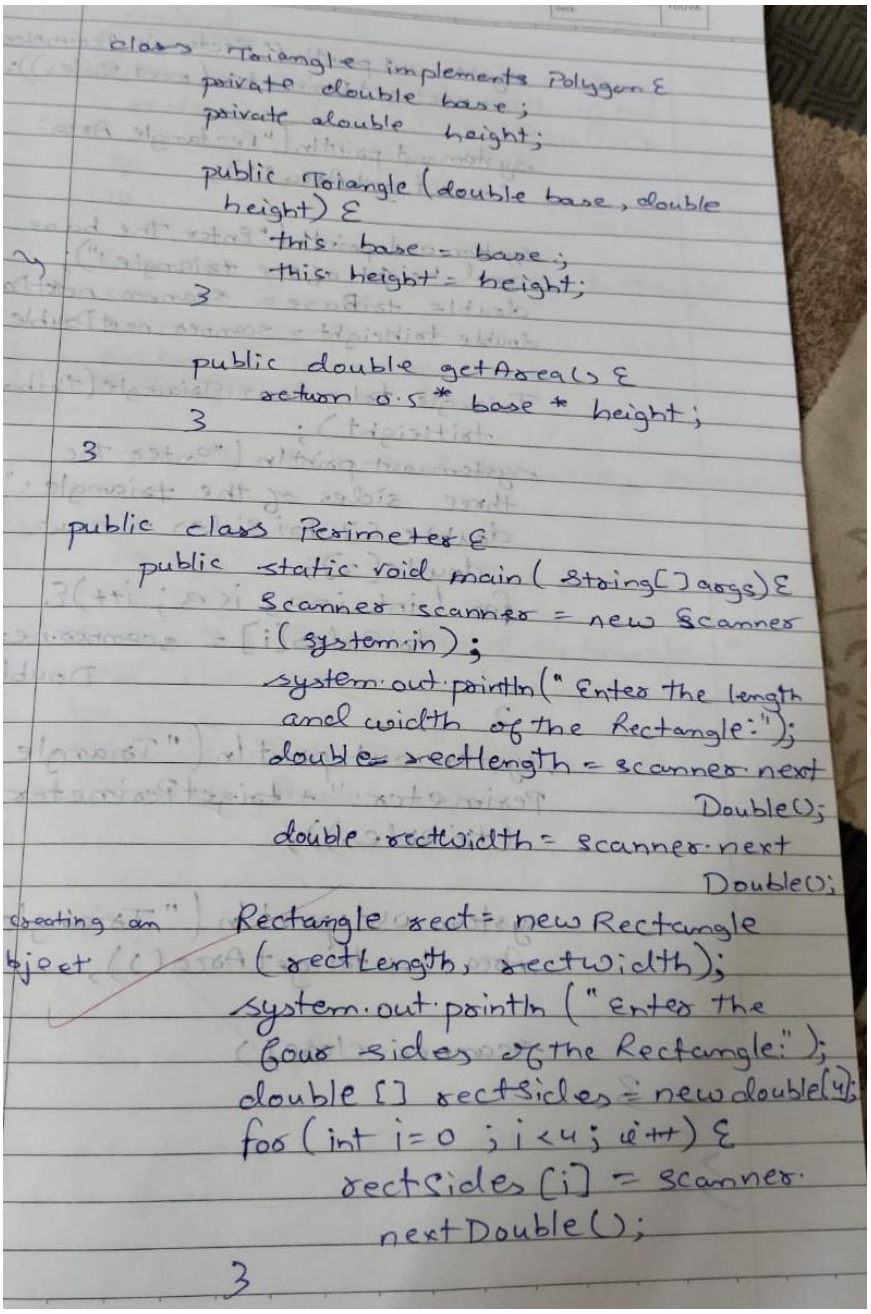
Output:



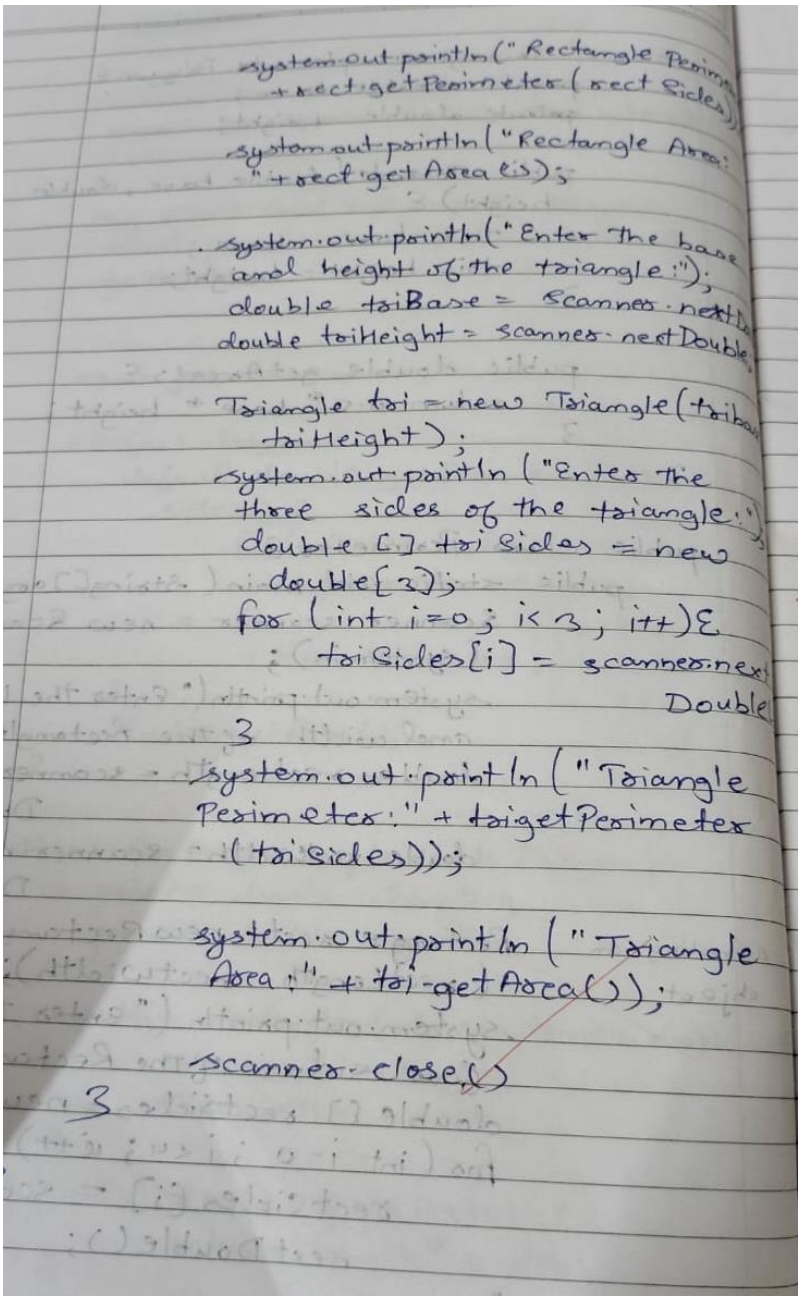
Program 8:
Interface
Algorithm:



```
impost java.util.scanner;

interface Polygon {
    default double getPerimeter(double[]
    sides) {
        for(double side : sides) {
            perimeter += side;
        }
        return perimeter;
    }
    double getArea();
}

class Rectangle implements Polygon {
    private double length;
    private double width;

    public Rectangle {double length, double wid
        This.Length = length;
        This.width = width;
    }
    public double getArea() {
        return length * width;
    }
}
```



```
class Triangle implements Polygon {
    private double base;
    private double height;

    public Triangle (double base, double
     height) {
        this.base = base;
        this.height = height;
    }

    public double getArea() {
        return 0.5 * base * height;
    }
}

public class Perimeter {
    public static void main (String[] args) {
        Scanner scanner = new Scanner
         (System.in);
        System.out.println("Enter the length
         and width of the Rectangle:");
        double rectLength = scanner.next
                                 Double();
        double rectWidth = scanner.next
                                 Double();
creating an  Rectangle rect = new Rectangle
object       (rectLength, rectWidth);
        System.out.println ("Enter the
         four sides of the Rectangle:");
        double [] rectSides = new double[4];
        for (int i=0 ; i<4; i++) {
            rectSides[i] = scanner.
             nextDouble();
        }
```



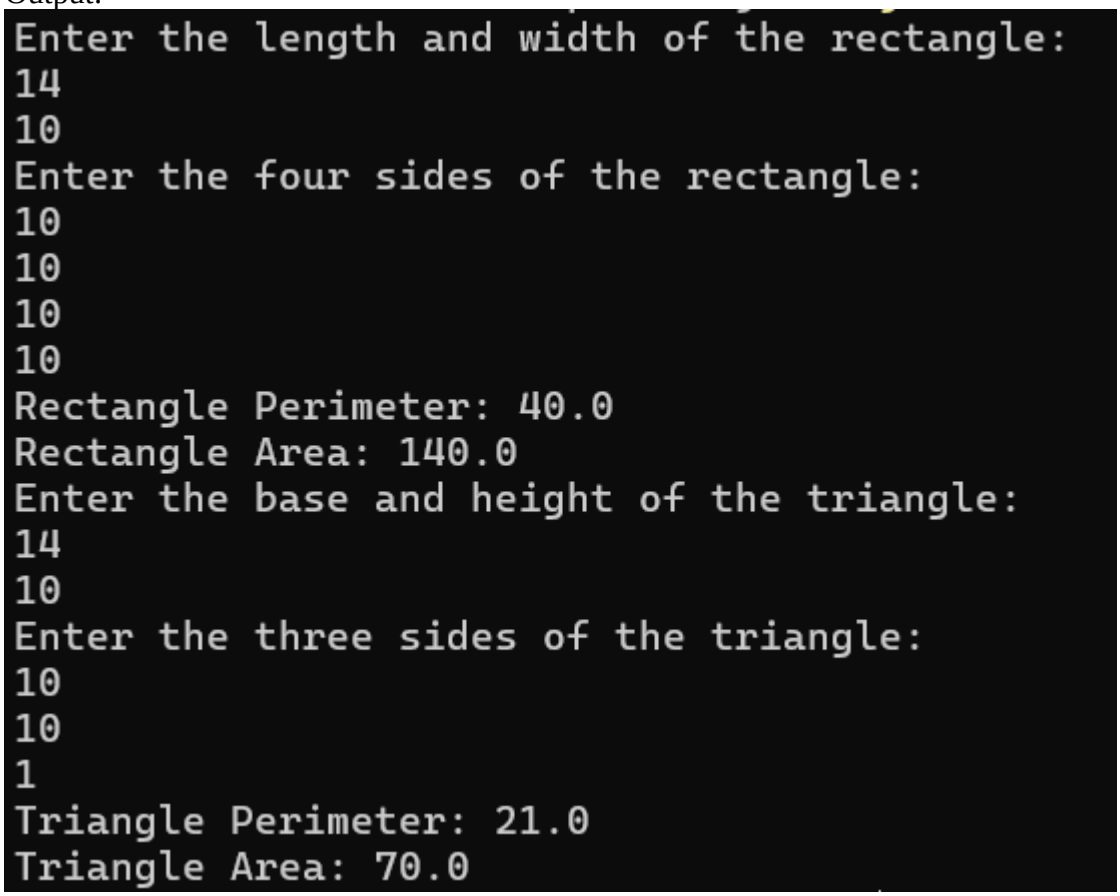
Code:

```
import java.util.Scanner;

interface Polygon {
    default double getPerimeter(double[] sides) {
        double perimeter = 0;
        for (double side : sides) {
            perimeter += side;
        }
        return perimeter;
    }
```

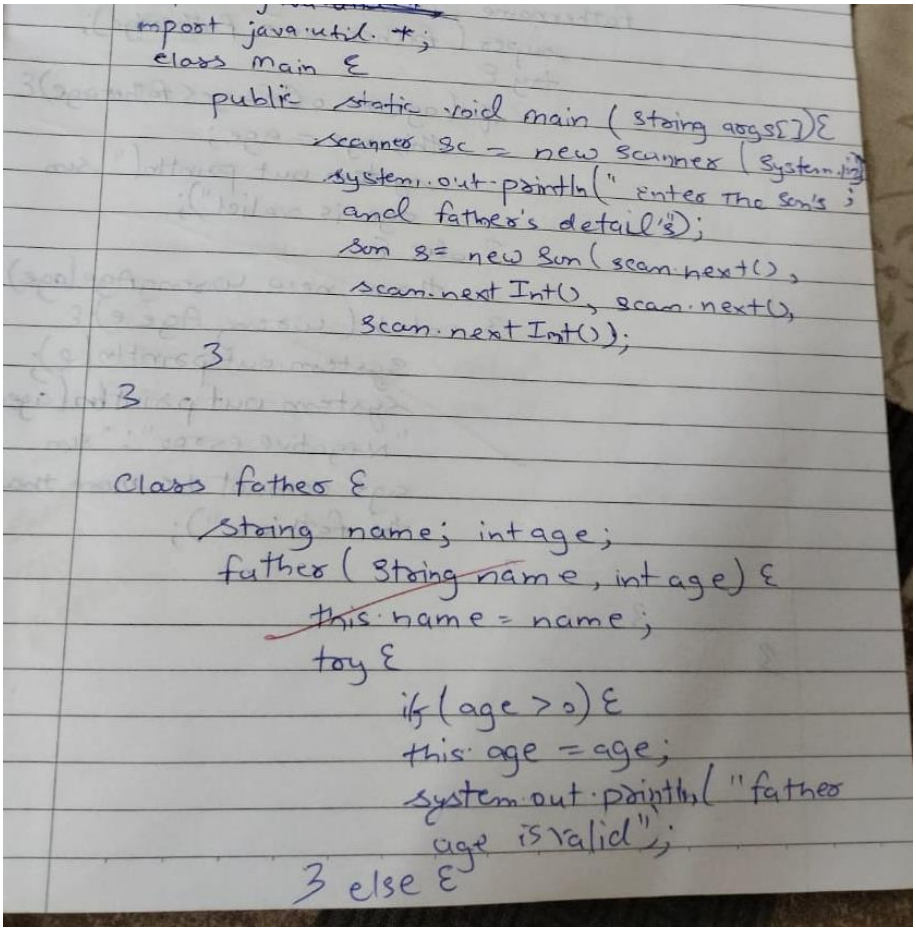
```
    double getArea();
}

class Rectangle implements Polygon {
    private double length;
    private double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    public double getArea() {
        return length * width;
    }
}

class Triangle implements Polygon {
    private double base;
    private double height;

    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }

    public double getArea() {
        return 0.5 * base * height;
    }
}

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

        System.out.println("Enter the length and width of the rectangle:");
        double rectLength = scanner.nextDouble();
        double rectWidth = scanner.nextDouble();

        Rectangle rect = new Rectangle(rectLength, rectWidth);
        System.out.println("Enter the four sides of the rectangle:");
        double[] rectSides = new double[4];
        for (int i = 0; i < 4; i++) {
            rectSides[i] = scanner.nextDouble();
        }
        System.out.println("Rectangle Perimeter: " + rect.getPerimeter(rectSides));
```

```
        System.out.println("Rectangle Area: " + rect.getArea());

        System.out.println("Enter the base and height of the triangle:");
        double triBase = scanner.nextDouble();
        double triHeight = scanner.nextDouble();

        Triangle tri = new Triangle(triBase, triHeight);
        System.out.println("Enter the three sides of the triangle:");
        double[] triSides = new double[3];
        for (int i = 0; i < 3; i++) {
            triSides[i] = scanner.nextDouble();
        }
        System.out.println("Triangle Perimeter: " + tri.getPerimeter(triSides));
        System.out.println("Triangle Area: " + tri.getArea());

        scanner.close();
    }
}
```

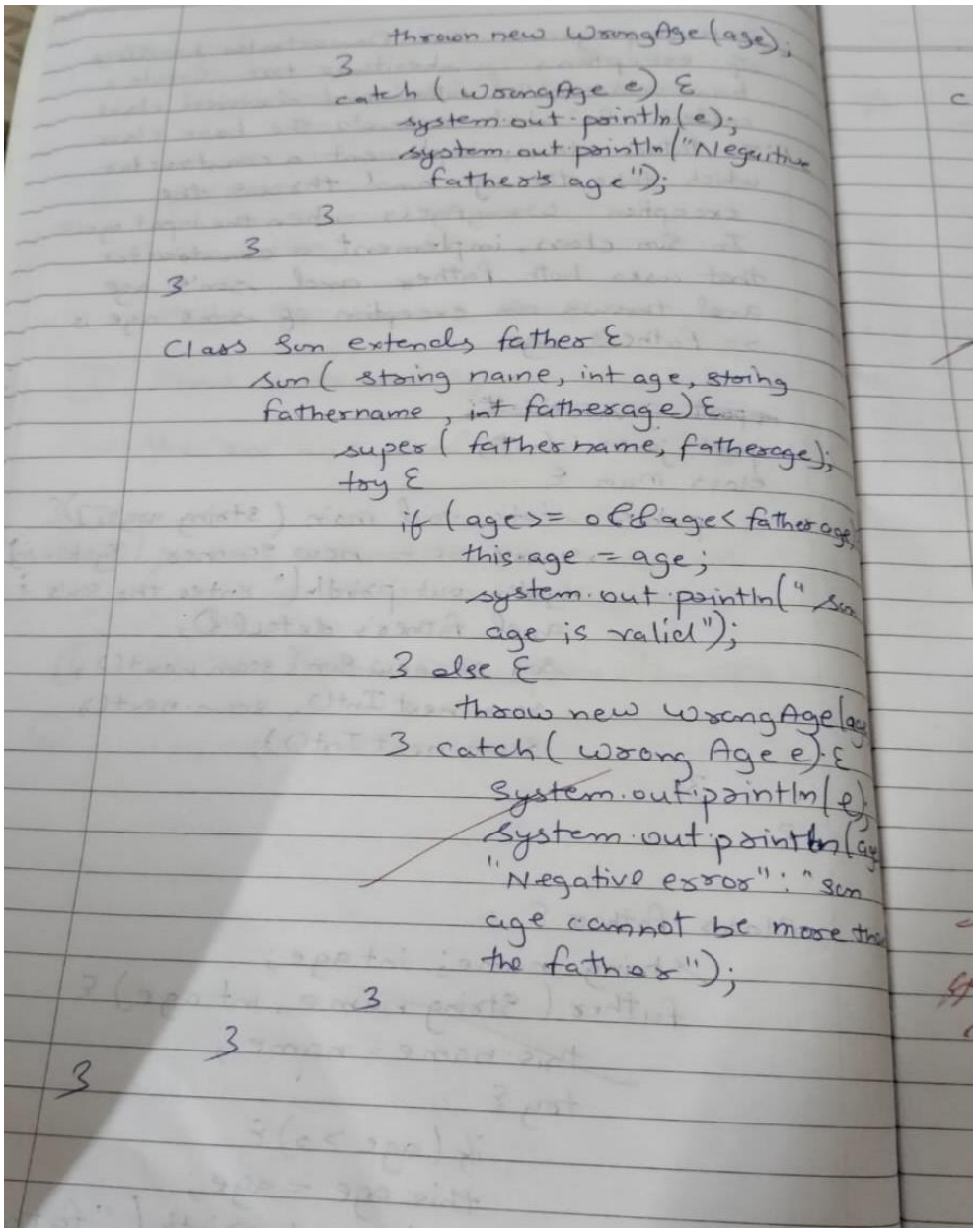
Output:



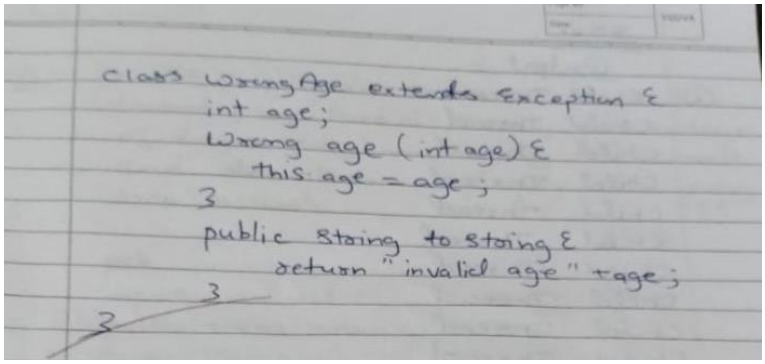
Program 9:
Father and Son Exception
Algorithm:



```
import java.util.*;
class main {
    public static void main (String args[]){
        Scanner sc = new Scanner (System.in);
        System.out.println(" enter the son's
        and father's detail's");
        Son s= new Son (scan.next(),
        scan.nextInt(), scan.next(),
        scan.nextInt());
    }
}

class father {
    String name; int age;
    father ( String name, int age) {
        this.name = name;
        try {
            if (age > 0) {
            this.age = age;
            System.out.println( "father
              age is valid");
        } else {
```



```
            throw new WrongAge(age);
        }
        catch (WrongAge e) {
            System.out.println(e);
            System.out.println("Negative
              father's age");
        }
    }
}

class Son extends father {
    Son( string name, int age, string
    fathername, int fatherage) {
        super( fathername, fatherage);
        try {
            if (age>= 0 && age< fatherage)
                this.age = age;
                system.out.println("Son
                  age is valid");
            } else {
                throw new WrongAge(age
            } catch( WrongAge e) {
                System.out.println(e);
                System.out.println(age
                "Negative error": "Son
                age cannot be more tha
                the father");
            }
        }
    }
}
```



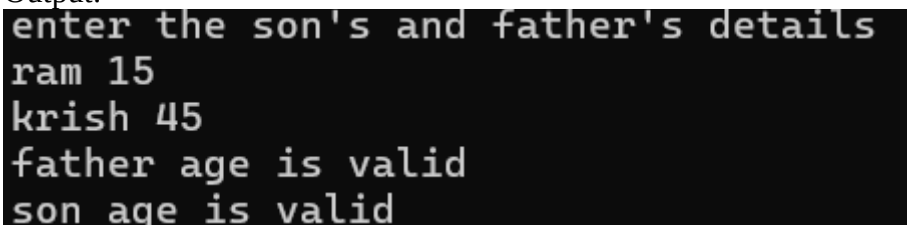
```
class WrongAge extends Exception {
    int age;
    Wrong age (int age) {
        this.age = age;
    }
    public String to string {
        return "invalid age" +age;
    }
}
```

Code:

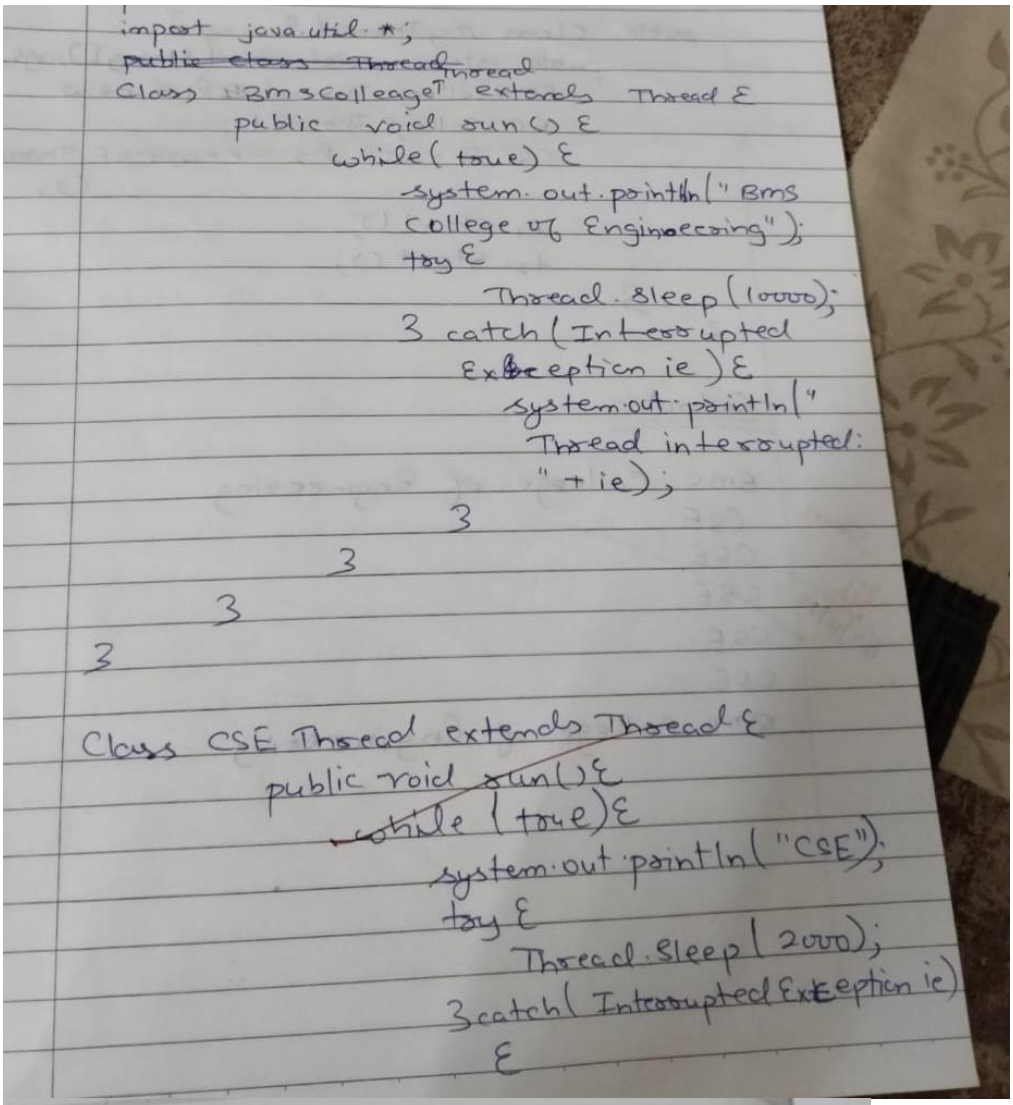
```
import java.util.*;
class Main{
public static void main(String args[]){
Scanner scan =new Scanner(System.in);
System.out.println("enter the son's and father's details");
Son s=new Son(scan.next(),scan.nextInt(),scan.next(),scan.nextInt());

}
}
class Father{
String name;int age;
Father(String name,int age){
this.name=name;
try{
if(age>0){
this.age=age;
System.out.println("father age is valid");
}else{
throw new WrongAge(age);
}
}catch(WrongAge e){
System.out.println(e);
System.out.println("negative fathers age");
}
}
}
class Son extends Father{
Son(String name,int age,String fathername,int fatherage){
super(fathername,fatherage);
try{
if(age>=0 && age<fatherage){
this.age=age;
System.out.println("son age is valid");
}else{
throw new WrongAge(age);
}
}catch(WrongAge e){
System.out.println(e);
System.out.println(age<0? "negative error" :"sons age cannot be more than the fathers");
}

}
}
class WrongAge extends Exception{
int age;
WrongAge(int age){
```

```
this.age=age;
}
public String toString(){
return "invalid age "+age;
}
}
```

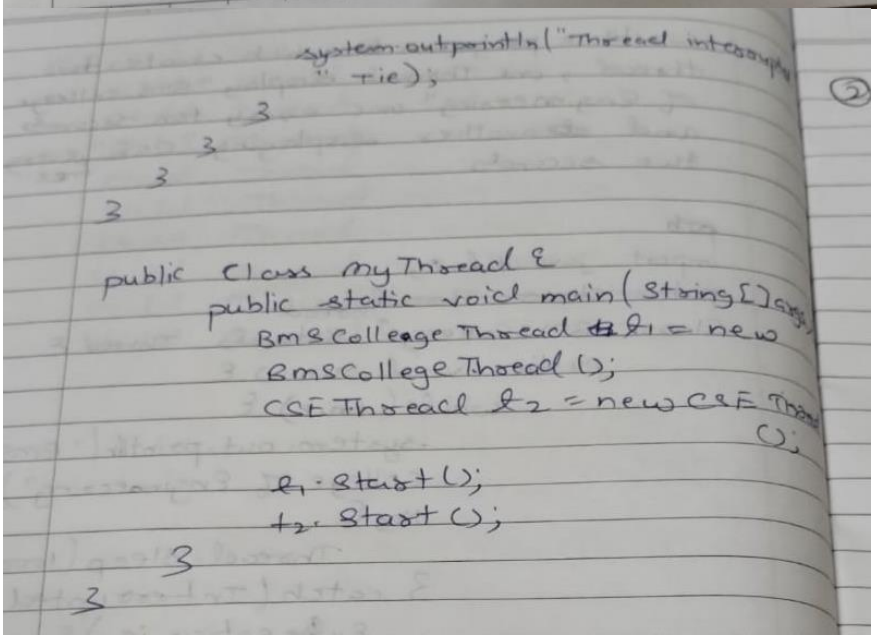
Output:



Program 10:
Thread Program
Algorithm:



```
import java.util.*;
public class Thread
Class BmsCollege extends Thread {
    public void run() {
        while(true) {
            System.out.println("Bms
            College of Engineering");
            try {
                Thread.sleep(10000);
            } catch(InterruptedException ie) {
                System.out.println("
                Thread interrupted:
                " + ie);
            }
        }
    }
}

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

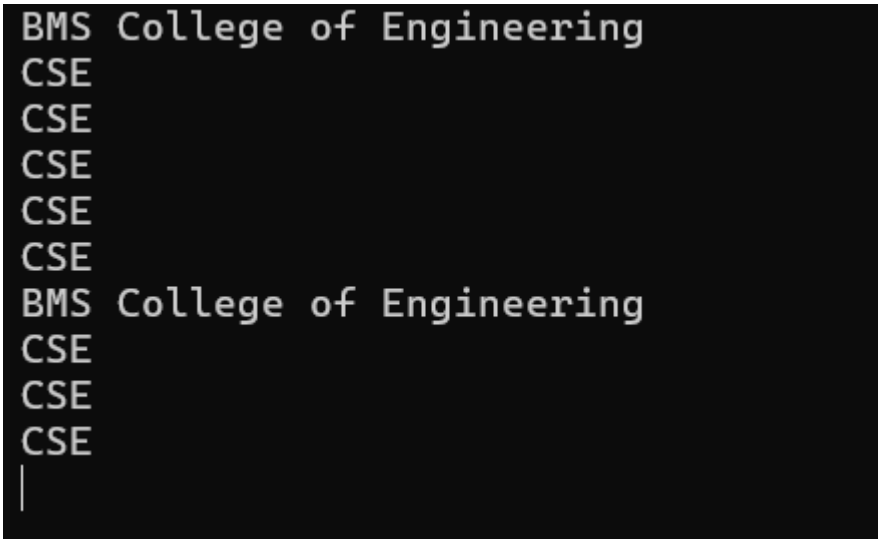


```
                System.out.println("Thread interrupted
                " + ie);
            }
        }
    }
}

public Class MyThread {
    public static void main(String[] args)
        BmsCollege Thread t1 = new
        BmsCollege Thread();
        CSE Thread t2 = new CSE Thread
                                    ();

        t1.start();
        t2.start();
    }
}
```

Code:

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

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

public class MyThread {
    public static void main(String[] args) {
        BMSCollegeThread t1 = new BMSCollegeThread();
        CSEThread t2 = new CSEThread();

        t1.start();
        t2.start();
    }
}
```

Output:


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