VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

B.Raja Simha Reddy (1BM23CS070)

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 **BOLLA RAJA SIMHA REDDY (1BM23CS070),** who is Bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object-Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Prof. Swathi Sridharan Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Sl. No.	Date	Experiment Title	Page No.
1	1/10/24	Quadratic Equation	4-7
2	8/10/24	Calculating SGPA	8-12
3	15/10/24	Book Details	13-17
4	22/10/24	Abstract Class Shape	18-21
5	29/10/24	Bank Details	21-26
6	12/11/24	Packages	27-33
7	19/11/24	Interface	33-37
8	26/11/24	Exception Handling	37 -40
9	3/12/24	Threads	41-43
10	3/12/24	GUI – Java Swing	43-51

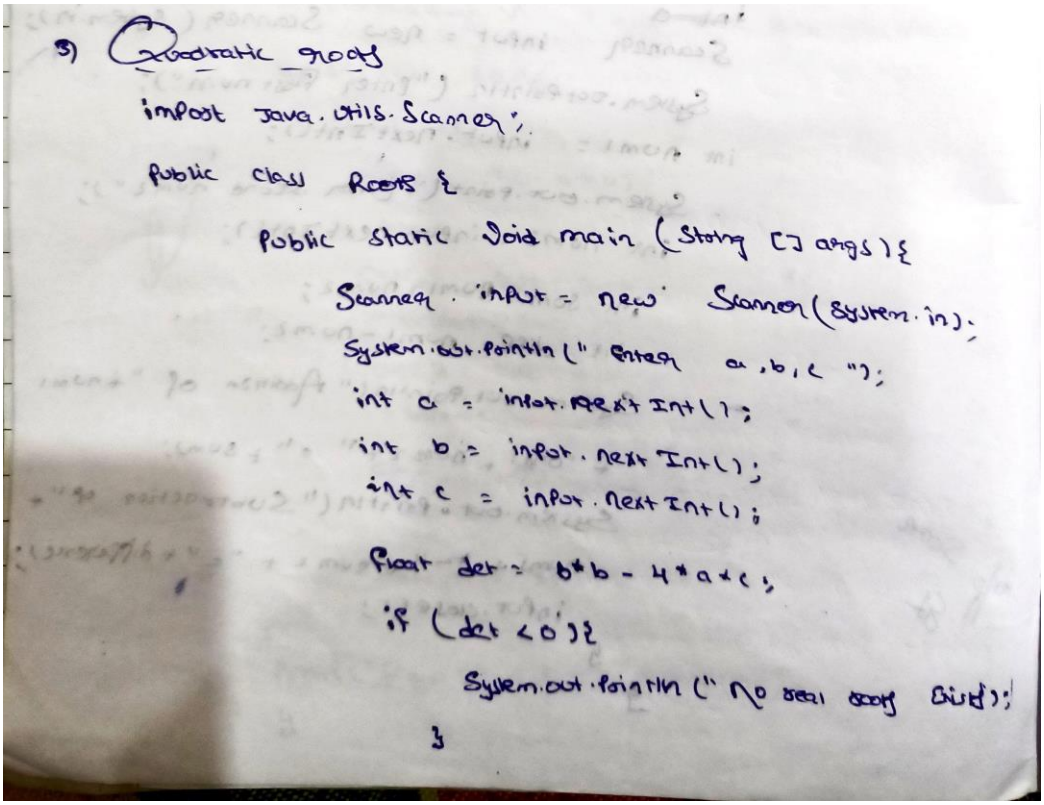
GitHub Link:
https://github.com/Rajasimhareddybolla/OOJ_LAB_U

Program 1

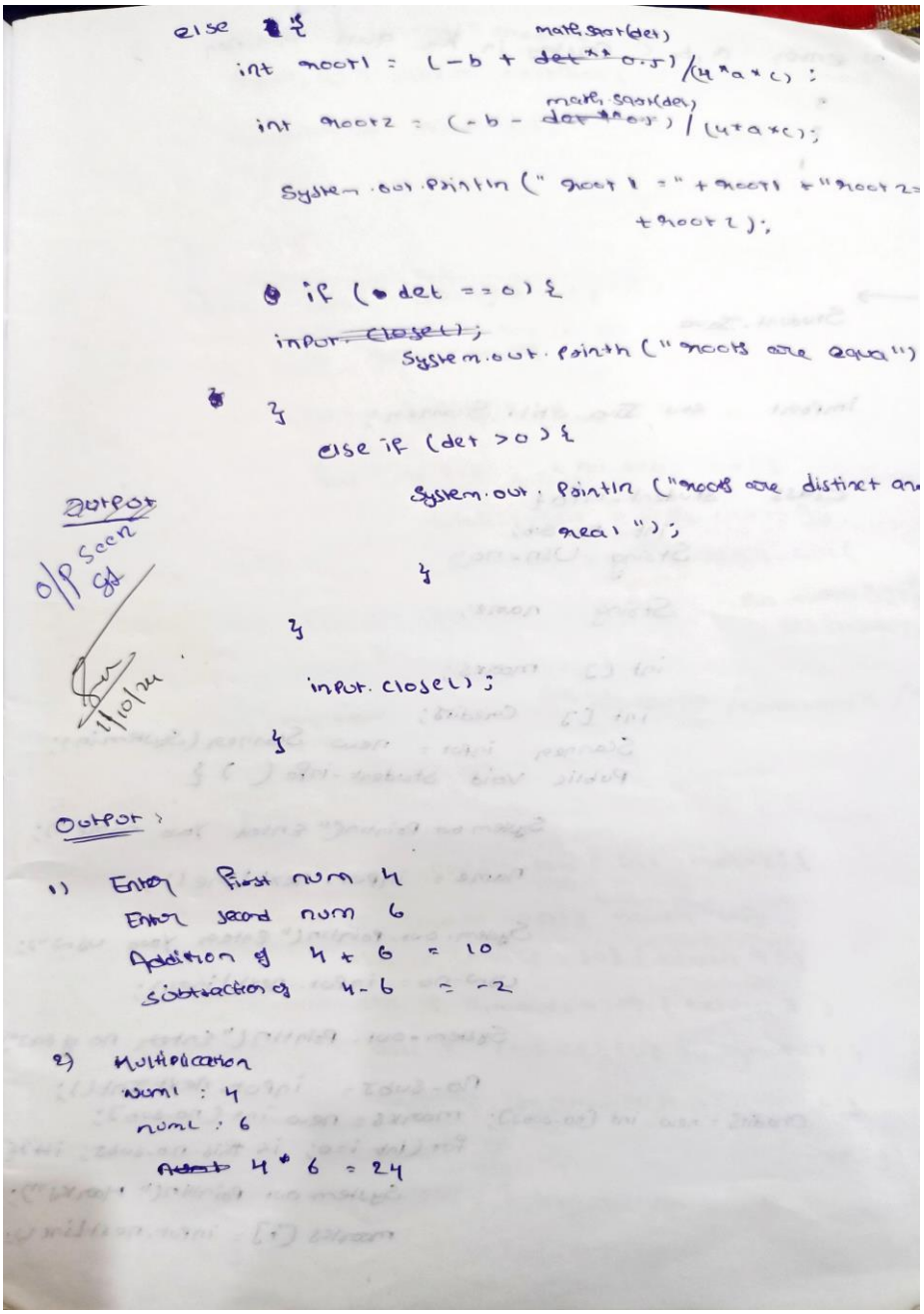
Implemente Quadratic Equation

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:



```
3) Quadratic_roots

import java.utils.Scanner;

public class Roots {

    Public static void main (String [] args){

        Scanner input = new Scanner(System.in);

        System.out.println(" Enter a,b,c ");

        int a = input.nextInt();

        int b = input.nextInt();

        int c = input.nextInt();

        float det = b*b - 4*a*c;

        if (det <0){

            System.out.println(" No real roots Exist");

        }
```

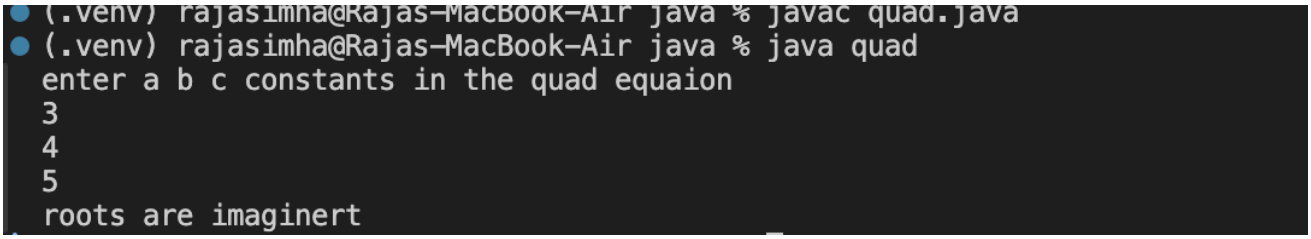


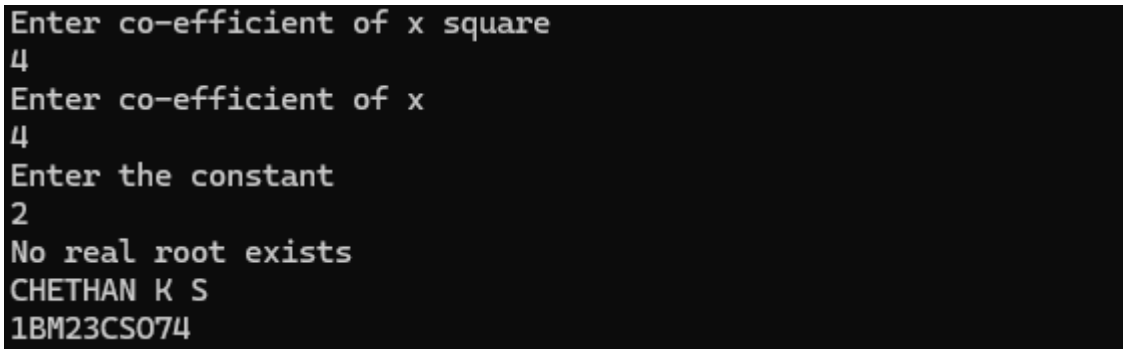
Code:

```
import java.util.Scanner;
public class quad{
	public static void main(String [] args){
		Scanner input = new Scanner(System.in);
		System.out.println("enter a b c constants in the quad equaion");
		int a = input.nextInt();
		int b = input.nextInt();
		int c = input.nextInt();
```

```
                float det = b*b -4*a*c;
                if (det < 0){
                        System.out.println("roots are imaginert ");
                        }
                else {
                        double root1 = (-b+Math.sqrt(det) )/ (2*a);
                        double root2 =( -b - Math.sqrt(det))/ (2*a);
                        if (det == 0){
                                System.out.println("roots are same ");
                                System.out.println("root 1 = " + root1 +"\n root2 -"+root2);
                        }
                        else{
                                System.out.println("roots are real and distinct ");
                                System.out.println("root 1 = " + root1 +"\n root2 -"+root2);
                        }

                        }
        }
}
```

Output:



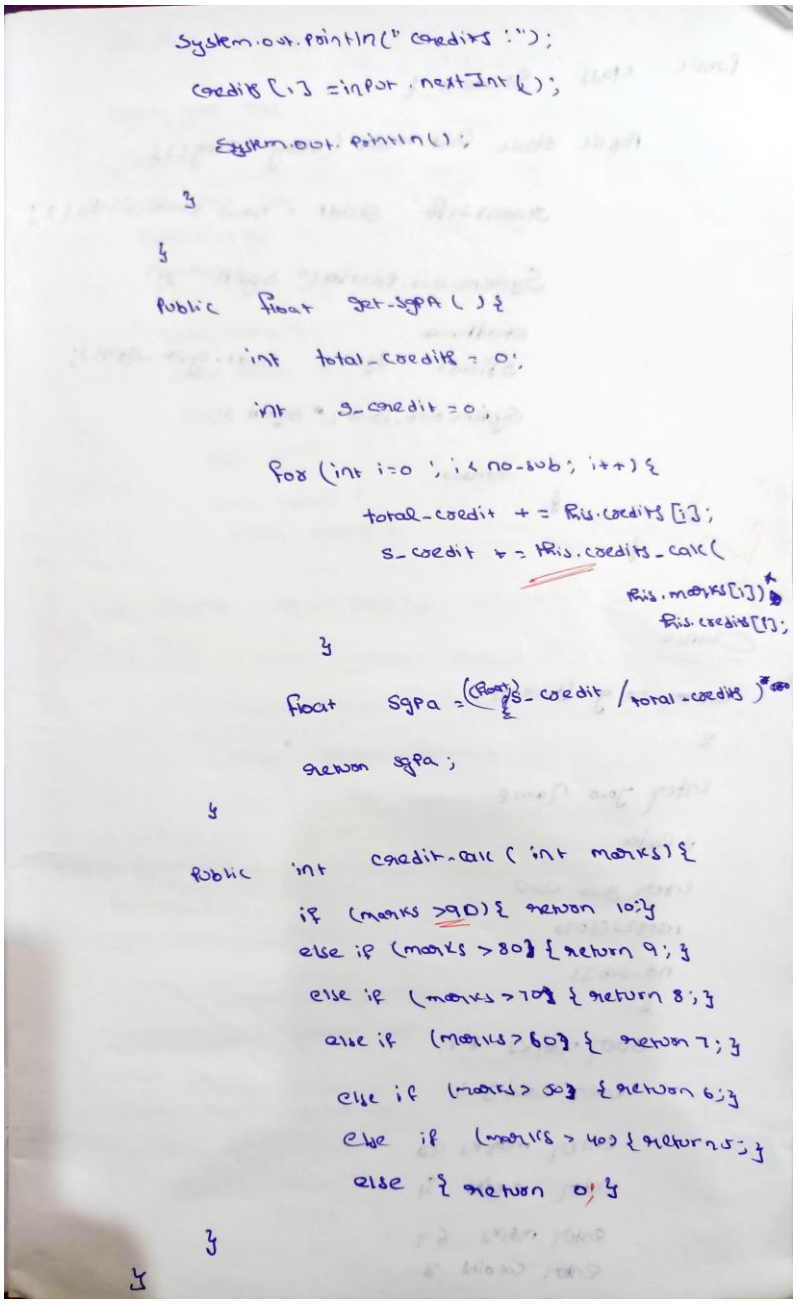


```
Enter co-efficient of x square
2
Enter co-efficient of x
4
Enter the constant
2
 The roots are equal -1.0
```

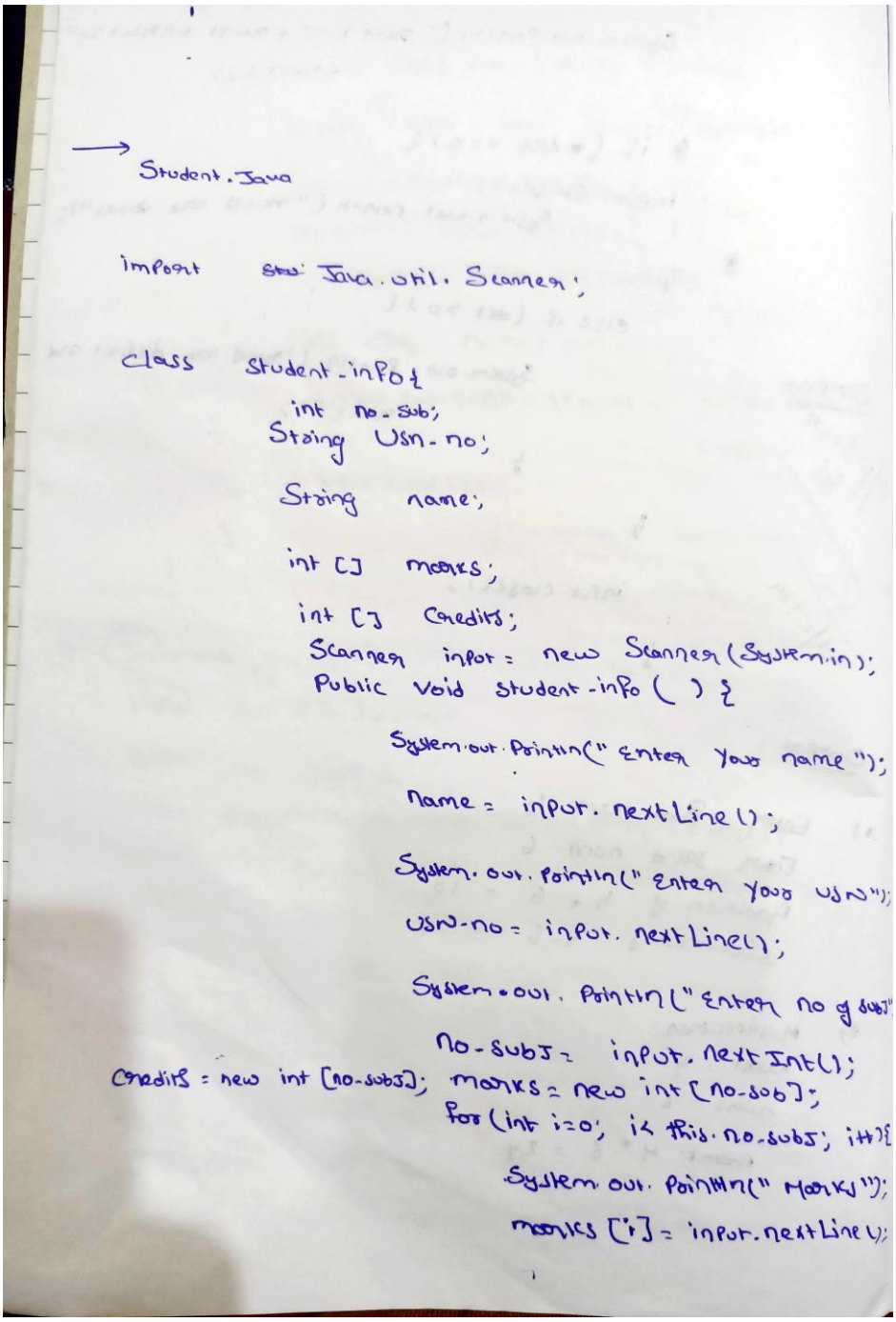
Program 2

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.

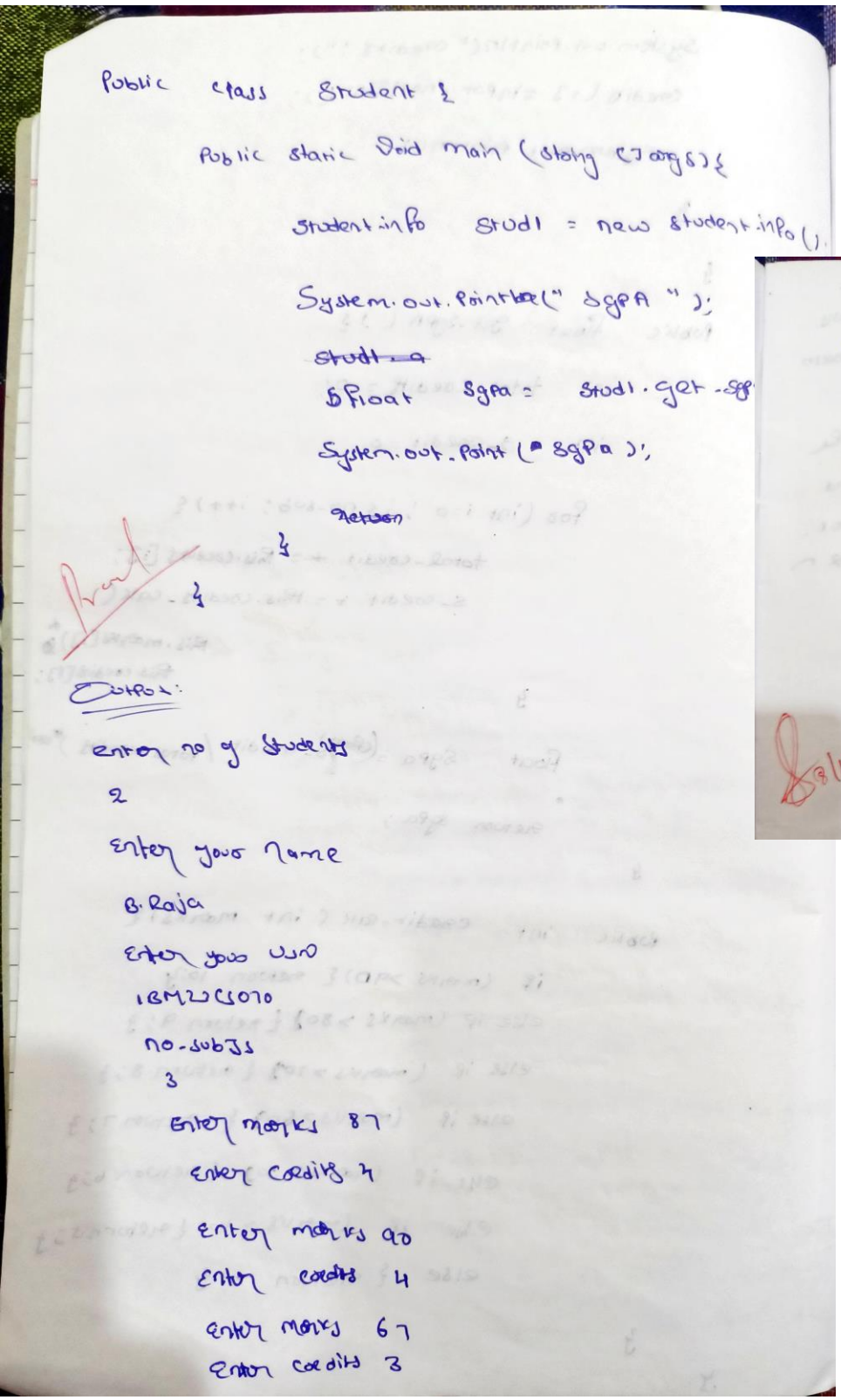


```
System.out.println(" credits :");
credits[i] = input.nextInt();
System.out.println();
}
}
public float get_SgPA() {
    int total_credits = 0;
    int s_credit = 0;
    for (int i=0 ; i< no_sub; i++) {
        total_credit += this.credits[i];
        s_credit += this.credits_calc(
                                this.marks[i])
                                * this.credits[i];
    }
    float sgpa = (float)s_credit / total_credits;
    return sgpa;
}
public int credit_calc ( 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; }
}
}
```



→ Student.Java

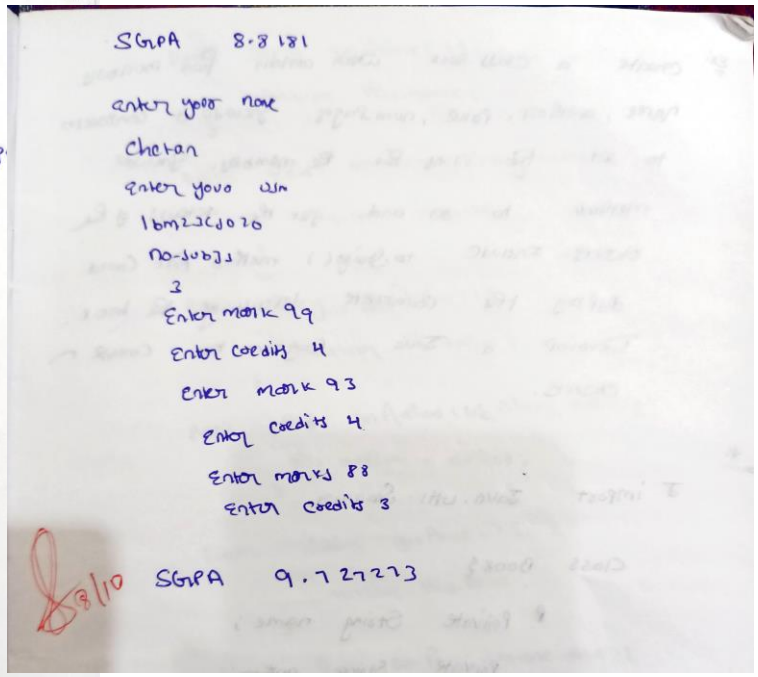
```
import java.util.Scanner;

class Student_info{
    int no_sub;
    String Usn_no;
    String name;
    int [] marks;
    int [] Credits;
    Scanner input = new Scanner(System.in);
    Public void student_info ( ) {
        System.out.Println(" Enter Your name ");
        name = input.nextLine();
        System.out.println(" Enter Your USN");
        USN_no = input.nextLine();
        System.out.Println(" Enter no of subj"
        no_subj = input.nextInt();
Credits = new int [no_subj]; marks = new int [no_sub];
        for (int i=0; i< this.no_subj; i++){
            System.out.Println(" Marks");
            marks [i] = input.nextLine();
```



```
Public class Student {
    Public static Void main (String [] args){
        Student info Stud1 = new Student info ();
        System.out.Println(" SGPA ");
        Stud1.a
        Float Sgpa = Stud1.get.sgp
        System.out.Print (" SgPa ");
        Return
    }
}
```

Output:

```
Enter no of students
2
Enter your name
B.Raja
Enter your usn
1BM21CS070
no-subjs
3
Enter marks 87
Enter credits 4
Enter marks 90
Enter credits 4
Enter marks 67
Enter credits 3
```



```
SGPA 8.8181
enter your name
Chetan
Enter your usn
1bm21cs070
no-subjs
3
Enter mark 99
Enter credits 4
Enter mark 93
Enter credits 4
Enter marks 88
Enter credits 3
8/10 SGPA 9.727273
```

```
import java.util.Scanner;

class student_info {
        String Usn_no;
        String name;
        int marks[];
        int credits[];
        int no_sub;
        Scanner input = new
Scanner(System.in);

        public void get_data() {
```

```
            System.out.println("Enter your name");
            name = input.nextLine();
            System.out.println("Enter your usn");
            Usn_no = input.nextLine();
            System.out.println("no_subjs");
            no_sub = input.nextInt();
            credits = new int[no_sub];
            marks = new int[no_sub];
            for (int i = 0; i < no_sub; i++) {
                  System.out.print("Enter Mark");
                  marks[i] = input.nextInt();
                  System.out.print("Enter Credits");
                  credits[i] = input.nextInt();
            }
            System.out.println();
      }

      public float get_sgpa() {

            int total_credits = 0;
            int s_grade = 0;
            for (int i = 0; i < no_sub; i++) {
                  total_credits += credits[i];
                  s_grade += credit_cal(marks[i]) * credits[i];
            }
            float sgpa = (float) s_grade / total_credits;
            return sgpa;
      }

      public int credit_cal(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;
            }
            return 0;
      }
}

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

```
                Scanner input = new Scanner(System.in);
                System.out.print("enter no of students ");
                int student_no = input.nextInt();
                for (int i = 0; i < student_no; i++) {
                        student_info stud = new student_info();
                        stud.get_data();
                        System.out.println("SGPA");
                        System.out.println(stud.get_sgpa());
                }
                System.out.println("B.RAJA SIMHA REDDY");
                System.out.println("1BM23CS070");
        }
}
```

Output:

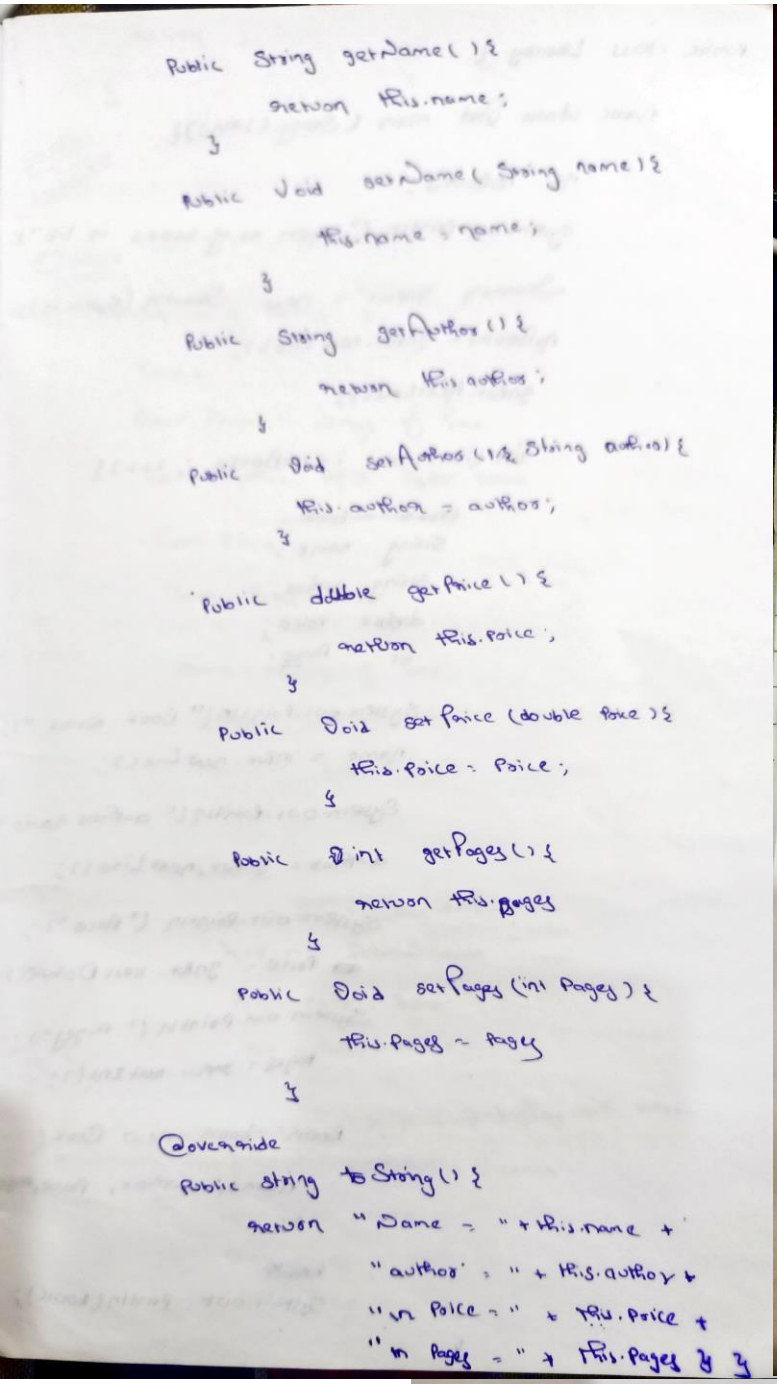
```
C:\Users\Admin\Desktop\raja_070>javac student.java

C:\Users\Admin\Desktop\raja_070>java student
enter no of students 3
Enter your name
raja
Enter your usn
1bm23cs070
no_subjs
2
Enter Mark99
Enter Credits3
Enter Mark87
Enter Credits4

SGPA
9.428572
Enter your name
chetan
Enter your usn
1bm23cs089
no_subjs
2
Enter Mark99
Enter Credits3
Enter Mark78
Enter Credits4
```

Program 3

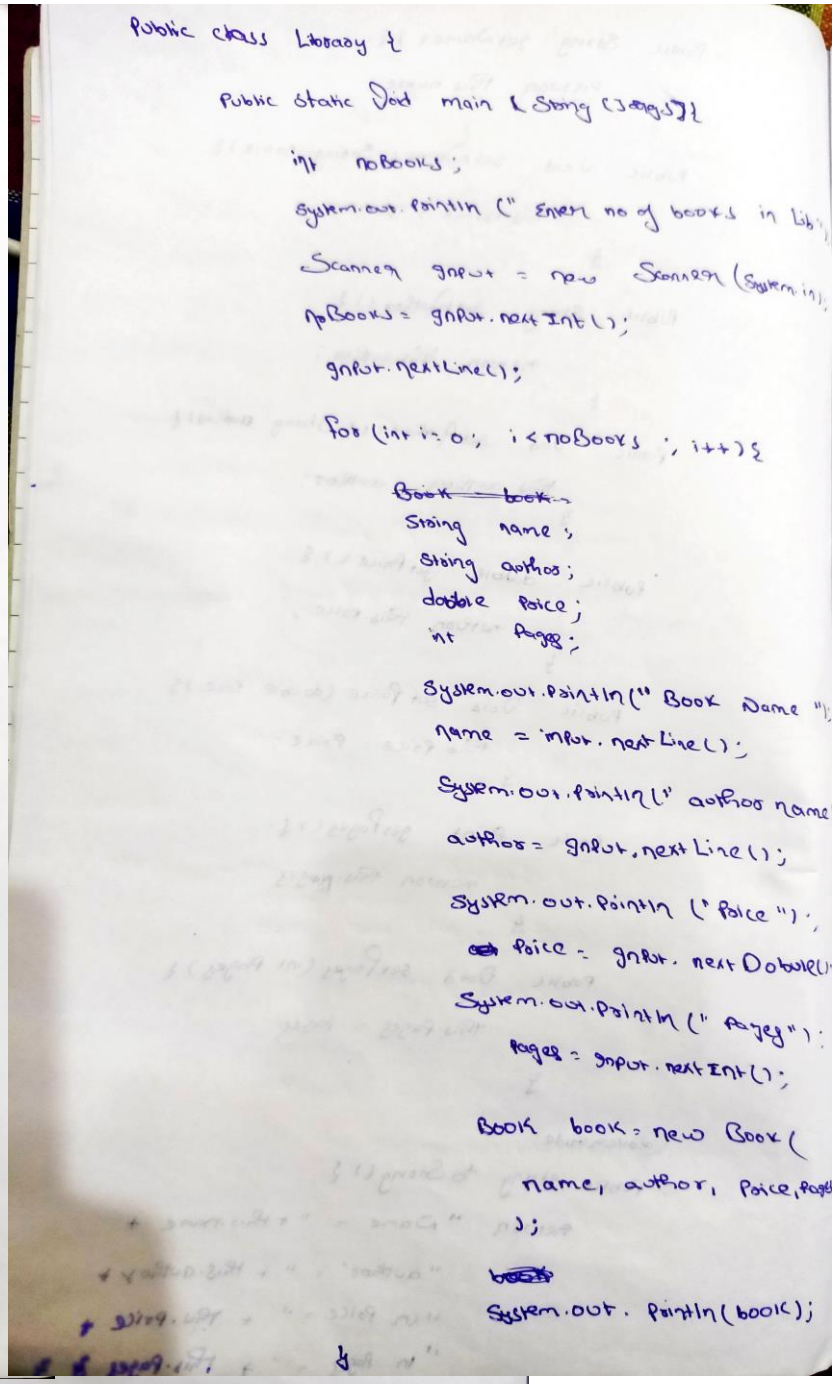
Book Details

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( ) that could display the complete details of the book. Develop a Java program to create n book objects.

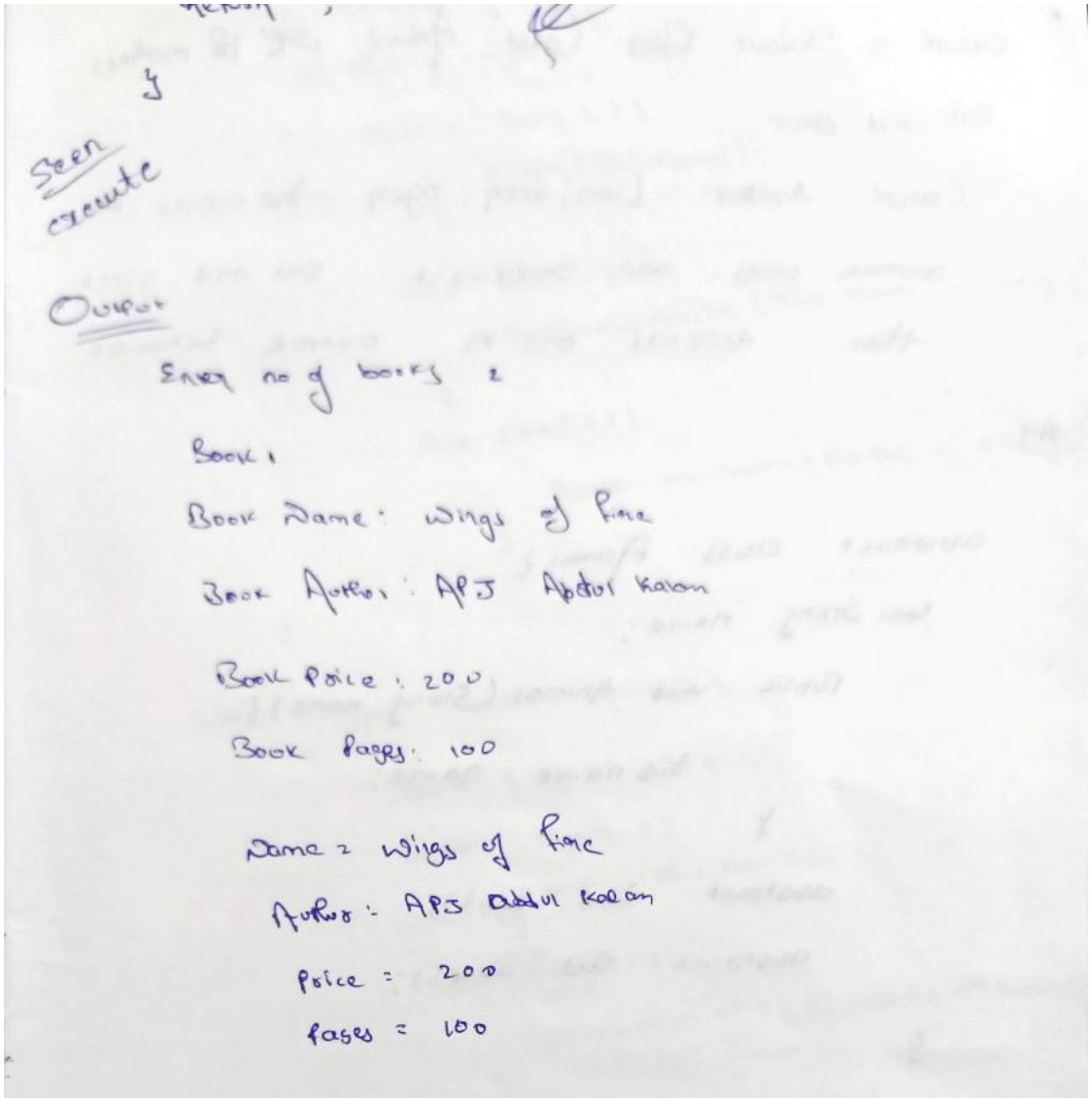
Algorithm:



```
Public String getName() {
    return this.name;
}
Public void setName(String name) {
    this.name = name;
}
Public String getAuthor() {
    return this.author;
}
Public void setAuthor(String author) {
    this.author = author;
}
Public double getPrice() {
    return this.price;
}
Public void setPrice(double price) {
    this.price = price;
}
Public int getPages() {
    return this.pages
}
Public void setPages(int pages) {
    this.pages = pages
}
@Override
Public String toString() {
    return "Name = " + this.name +
    "author = " + this.author +
    "\n Price = " + this.price +
    "\n Pages = " + this.pages }
```



```
Public class Library {
    Public static void main (String[] args){
        int noBooks;
        System.out.println("Enter no of books in Lib");
        Scanner input = new Scanner(System.in);
        noBooks = input.nextInt();
        input.nextLine();
        for (int i=0; i<noBooks; i++){
            String name;
            String author;
            double price;
            int pages;
            System.out.println("Book Name");
            name = input.nextLine();
            System.out.println("author name");
            author = input.nextLine();
            System.out.println("Price");
            price = input.nextDouble();
            System.out.println("pages");
            pages = input.nextInt();
            Book book = new Book(
                name, author, price, pages
            );
            System.out.println(book);
        }
```



Code:

```
import java.util.Scanner;

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

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

```
		this.price = price;
		this.pages = pages;
	}

	public String getName() {
		return this.name;
	}

	public String getAuthor() {
		return this.author;
	}

	public double getPrice() {
		return this.price;
	}

	public double getPages() {
		return this.pages;
	}

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

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

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

	public void setPages(int pages) {
		this.pages = pages;
	}

	@Override
	public String toString() {
		return "Name =" + this.name +
				"\nAuthor =" + this.author +
				"\nprice =" + this.price +
				"\npages =" + this.pages;
	}

}

public class Library {
	public static void main(String[] args) {
		Scanner input = new Scanner(System.in);
		System.out.print("enter no of Books ");
```

```
            int noBooks = input.nextInt();
            input.nextLine();
            for (int i = 0; i < noBooks; i++) {

                System.out.println("Book :" + (i + 1));
                String name;
                String author;
                double price;
                int pages;

                System.out.print("Book Name :");
                name = input.nextLine();
                System.out.print("Author Name :");
                author = input.nextLine();
                System.out.print("Book Price :");
                price = input.nextDouble();
                System.out.print("Book Pages :");
                pages = input.nextInt();


                Book book = new Book(name, author, price, pages);
                System.out.println();
                System.out.println(book);
                System.out.println();
                input.nextLine();

            }
            System.out.println("B.RAJA SIMHA REDDY");
            System.out.println("1BM23CS070");

        }
}
```

Output:

C:\Windows\System32\cmd.exe

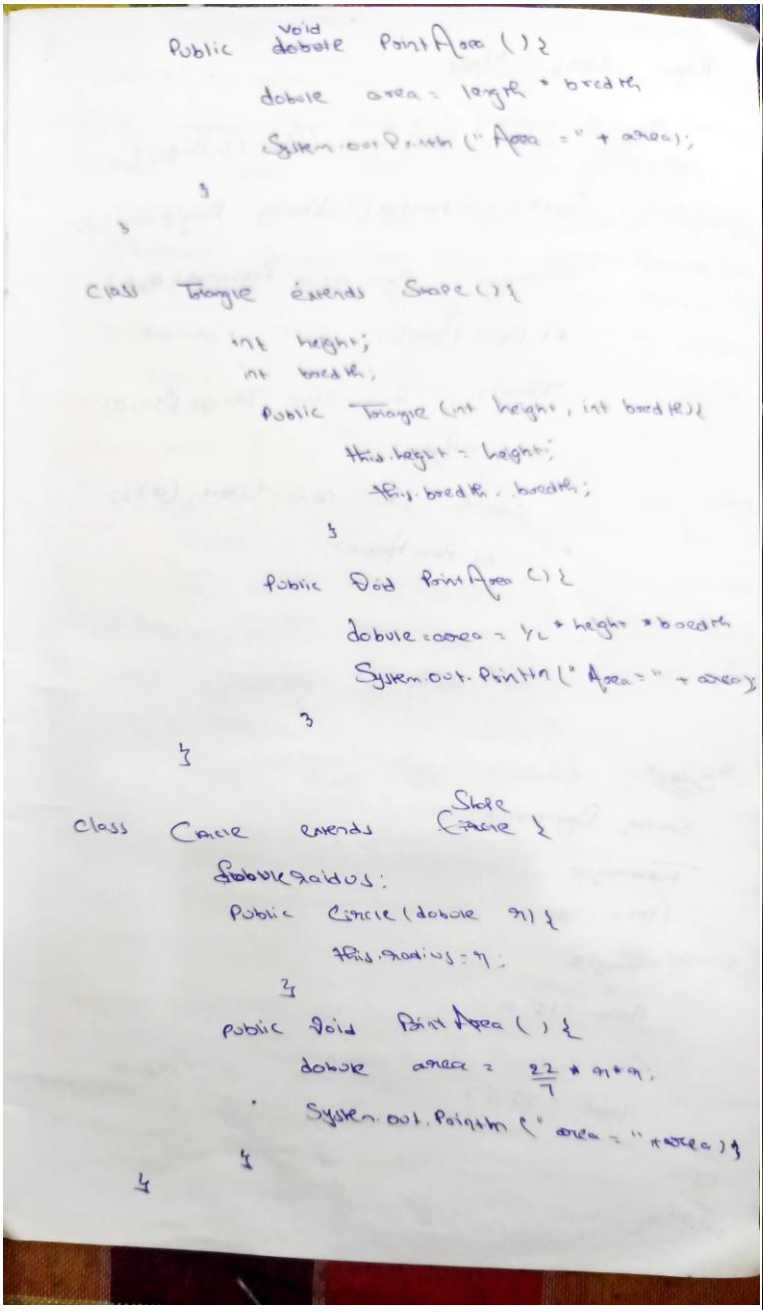
```
C:\Users\Admin\Desktop\raja_070>javac Library.java

C:\Users\Admin\Desktop\raja_070>java Library
enter no of Books 2
Book :1
Book Name :wings on fire
Author Name :apj abdul kalam
Book Price :200
Book Pages :100

Name =wings on fire
Author =apj abdul kalam
price =200.0
pages =100

Book :2
Book Name :tuesday with morry
Author Name :robhin sherma
Book Price :200
Book Pages :100

Name =tuesday with morry
Author =robhin sherma
price =200.0
pages =100

B.RAJA SIMHA REDDY
1BM23CS070
```

Program 4

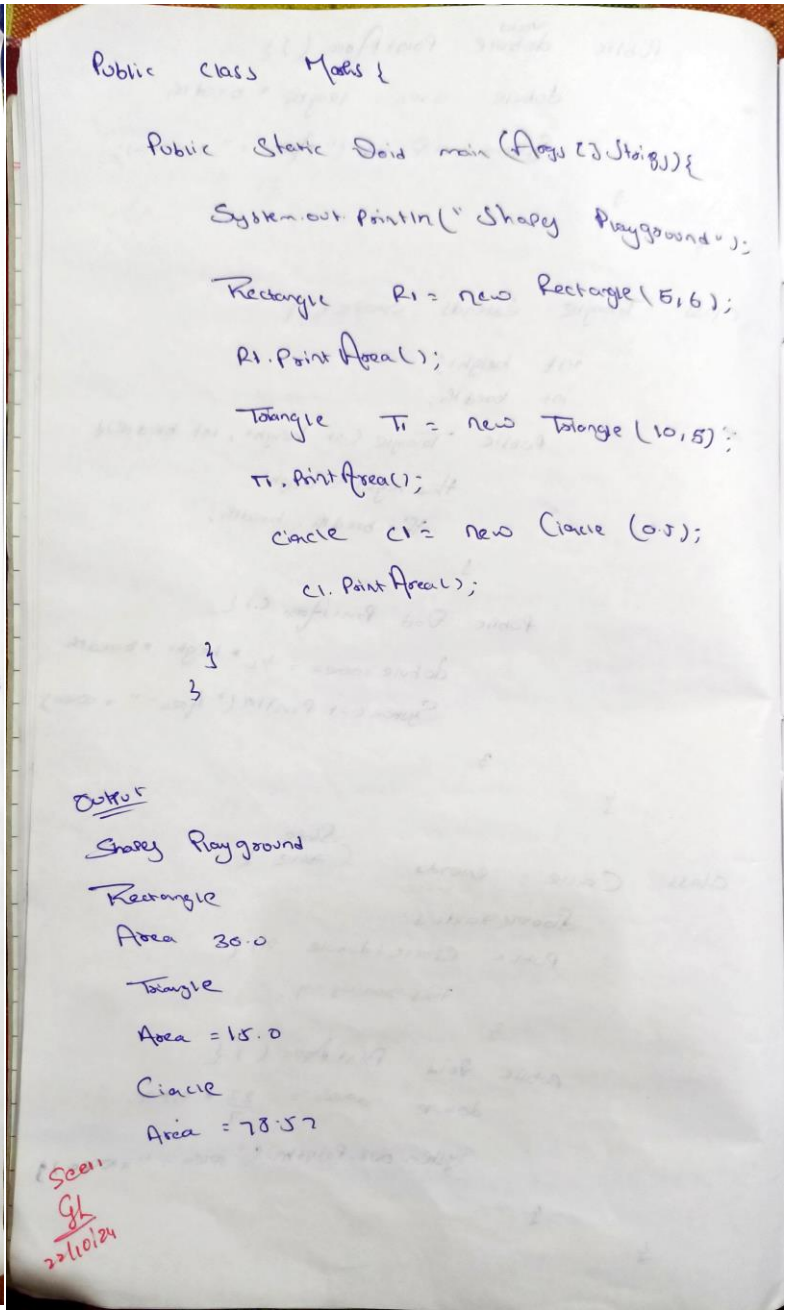
Abstract Class Shape

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:



```
Public void doble PrintArea (){
    doble area = langth * bredth
    System.out.println(" Area = " + area);
}
}

class Triangle extends Shape (){
    int height;
    int bredth;
    Public Triangle (int height, int bredth){
        this.height = height;
        this.bredth = bredth;
    }
    Public Void PrintArea (){
        doble area = 1/2 * height * bredth
        System.out.println(" Area = " + area);
    }
}

class Circle extends Shape Circle {
    doble raidus;
    Public Circle (doble r){
        this.raidus = r;
    }
    public Void PrintArea (){
        doble area = 22/7 * r*r;
        System.out.println(" area = " + area);
    }
}
```



```
Public class Maths {
    Public Static Void main (Args[] String){
        System.out.println(" Shapes Playground ");
        Rectangle R1 = new Rectangle (5,6);
        R1.PrintArea();
        Triangle T1 = new Triangle (10,5);
        T1.PrintArea();
        Circle C1 = new Circle (0.5);
        C1.PrintArea();
    }
}
```

Output

```
Shapes Playground
Rectangle
Area 30.0
Triangle
Area = 15.0
Circle
Area = 78.57
```

Seen
GL
22/10/24

Code:

```
import java.util.Scanner;

abstract class Shape {
    int a, b;

    abstract double printArea();

    void Value() {
        Scanner s = new Scanner(System.in);
```

```
        a = s.nextInt();
        b = s.nextInt();
    }

    void Valuerad() {
        Scanner s = new Scanner(System.in);
        a = s.nextInt();
    }
}

class Rectangle extends Shape {
    double printArea() {
        return a * b;
    }
}

class Triangle extends Shape {
    double printArea() {
        return 0.5 * a * b;
    }
}

class Circle extends Shape {
    double printArea() {
        return 3.14 * a * a;
    }
}

class shapeArea {
    public static void main(String args[]) {
        Rectangle r = new Rectangle();
        System.out.println("Enter the values of length and breadth: ");
        r.Value();
        Triangle t = new Triangle();
        System.out.println("Enter the values of base and height: ");
        t.Value();
        Circle c = new Circle();
        System.out.println("Enter the value of radius: ");
        c.Valuerad();
        System.out.println("The area of rectangle is: " + r.printArea());
        System.out.println("The area of Triangle is: " + t.printArea());
        System.out.println("The area of Circle is: " + c.printArea());

    }
```

Output :

}

```
rajasimha@Rajas-MacBook-Air 4_abstract % java shapeArea
Enter the values of length and breadth:
5
6
Enter the values of base and height:
6
7
Enter the value of radius:
9
The area of rectangle is: 30.0
The area of Triangle is: 21.0
The area of Circle is: 254.34
```

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.

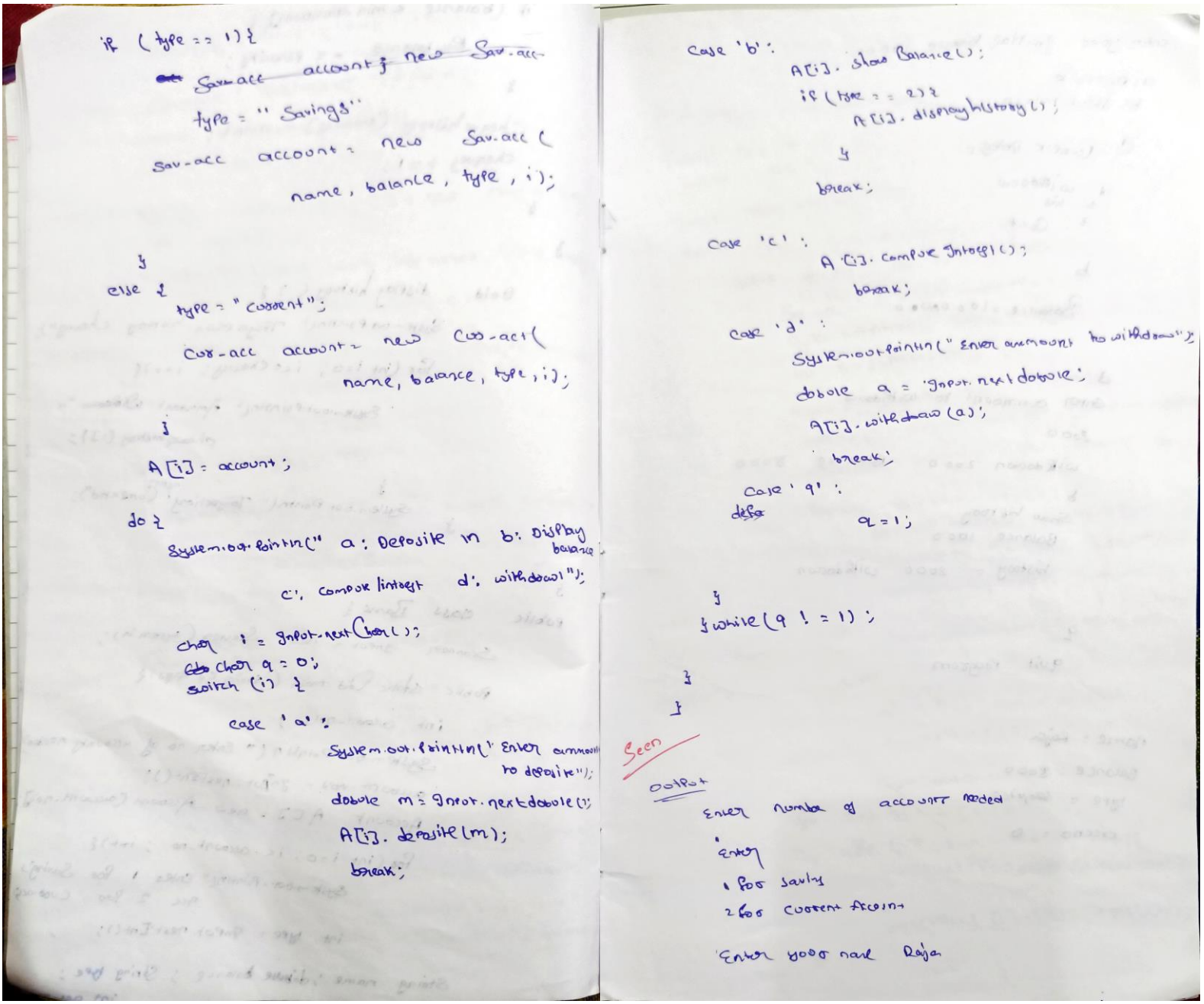
Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

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:

```
Class Cheque {

Class Cur-act implements Account {
    double min_amnt = 1000.0 ;
    double penality = 100.0 ;
    double cheque history [] = new double [100];
    int cheque = 0;
    Cur-act (String name, int accno, String acctype,
             double balance) {

        super( name, accno, acctype, balance);

    }

    Void ComposeInterest () {
        System.out.Println(" For current account no
            interest can be Composed");

    }

    Void withdraw (double amnt) {
        if (amnt > this.balance) {
            System.out.println(" Insufficient
                balance");
        }
        else {
            this.balance = this.balance - amnt;
            System.out.println(" withdrawn
                ammount" + amnt + "
                Balance = " + balance);
            if (balance < min_ammount) {
                this.balance -= penality;
            }
            cheque history [cheque] = amnt;
            cheque += 1;
        }
    }

    Void display history ( ) {
        System.out.println(" Transaction history cheque");
        For (int i=0; i<= cheque; i++) {
            System.out.println(" Ammount Withdrawn " +
                cheque history [i]);
        }
        System.out.println(" Transaction after Completed");
    }
}
public class Bank {
    Scanner input = new Scanner (System.in);
    Public static Void main (String [] args) {
        int account_no's ;
        System.out.println (" Enter no of accounts needed");
        account_no's = input.nextInt();
        Account A [] = new Account [account_no's];
        For (int i=0; i< account_no ; i++) {
            System.out.println(" Enter 1 For Saving
                Acc 2 For Curr-acc");
            int type = input.nextInt();
        String name ; double balance ; String type ;
                int acc_no ;
```



Code:

```
import java.util.*;

abstract class Account {
    String customerName;
    int accountNumber;
    double balance;
```

```
    String accountType;

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

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

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

    abstract void computeInterest();

    abstract void withdraw(double amount);
}

class SavAcct extends Account {
    final double interestRate = 0.04;

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

    @Override
    void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added. New balance: " + balance);
    }

    @Override
    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}

class CurAcct extends Account {
    double minBalance = 1000.00;
```

```
    double charge = 50.00;
    double[] chequeTransactions = new double[100];
    int chequeId = 0;

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

    @Override
    void computeInterest() {
        System.out.println("Interest cannot be calculated for a Current Account.");
    }

    @Override
    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            if (balance >= minBalance) {
                System.out.println("The updated balance is: " + balance);
            } else {
                balance -= charge;
                System.out.println("Penalty of 50.0 has been deducted. The new balance is: " + balance);
            }
            chequeTransactions[chequeId] = amount;
            chequeId++;
        } else {
            System.out.println("Insufficient balance. The withdrawal amount is greater than balance.");
        }
    }

    void displayTransactions() {
        for (int i = 0; i < chequeId; i++) {
            System.out.println("Transaction " + (i + 1) + ": " + chequeTransactions[i]);
        }
    }
}

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

        System.out.println("Enter account type:");
        System.out.println("1. Savings");
        System.out.println("2. Current");
        int choice = input.nextInt();
        input.nextLine();

        System.out.println("Enter your name:");
        String name = input.nextLine();
```

```
System.out.println("Enter your account number:");
int accountNumber = input.nextInt();

System.out.println("Enter the initial balance:");
double balance = input.nextDouble();

Account account;

if (choice == 1) {
   account = new SavAcct(name, accountNumber, balance);
} else {
   account = new CurAcct(name, accountNumber, balance);
}

int exit = 0;

while (exit != 1) {
   System.out.println("\nEnter the function to be done:");
   System.out.println("1. Deposit");
   System.out.println("2. Display balance");
   System.out.println("3. Compute and deposit interest");
   System.out.println("4. Withdrawal");
   System.out.println("5. Exit");

   int func = input.nextInt();

   switch (func) {
      case 1:
         System.out.println("Enter deposit amount:");
         double depAmount = input.nextDouble();
         account.deposit(depAmount);
         break;

      case 2:
         account.displayBalance();
         break;

      case 3:
         account.computeInterest();
         break;

      case 4:
         System.out.println("Enter withdrawal amount:");
         double withdrawAmount = input.nextDouble();
         account.withdraw(withdrawAmount);
         break;

      case 5:
         exit = 1;
         System.out.println("Exiting");
```

```
Balance = 102000.0
a: Deposit
b: Show Balance
c: Compute Interest
d: Withdraw
q: Quit
q
Enter:
1 for Savings
2 for Current Account
2
Enter your name:
murali
Enter your initial balance:
1000000
a: Deposit
b: Show Balance
c: Compute Interest
d: Withdraw
q: Quit
d
Enter amount to withdraw:
60000
Withdrawn: 60000.0
Current balance: 940000.0
a: Deposit
b: Show Balance
c: Compute Interest
d: Withdraw
q: Quit
```

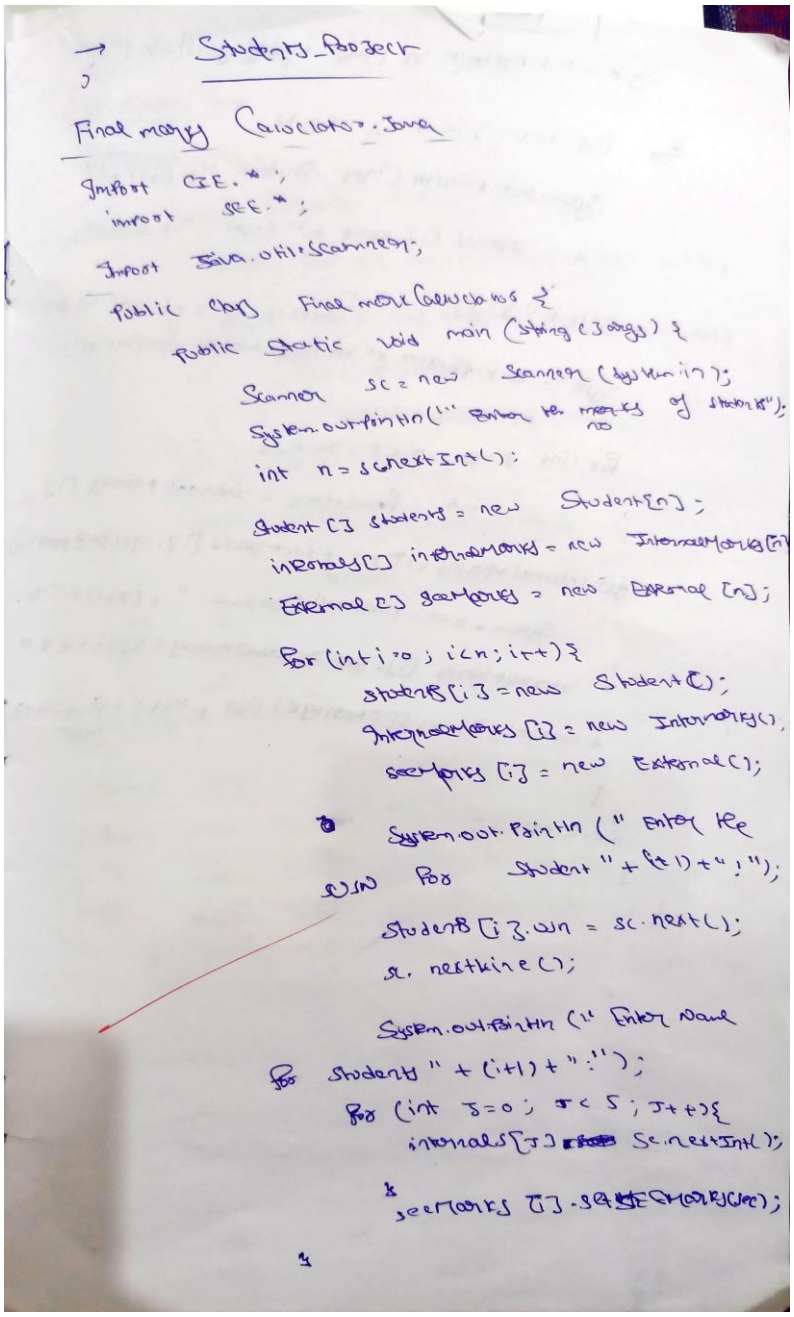
```
                break;

            default:
                System.out.println("Invalid input");
        }

        if (choice == 2) {
            ((CurAcct)
account).displayTransactions();
        }
    }
    input.close();
  }
}
```

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:



Student_Project

Final marks Calculator.java

```
import CIE.*;
import SEE.*;
import java.util.Scanner;
public class FinalMarksCalculator {
    public static void main (String[] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println(" Enter the marks of students");
        int n = sc.nextInt();
        Student[] students = new Student[n];
        Internals[] internalMarks = new Internals[n];
        External[] seeMarks = new External[n];
        for (int i=0; i<n; i++){
            students[i] = new Student();
            internalMarks[i] = new Internals();
            seeMarks[i] = new External();
            System.out.println (" Enter the
usn for Student " + (i+1) + ":");
            students[i].usn = sc.next();
            sc.nextLine();
            System.out.println (" Enter Name
for Student " + (i+1) + ":");
            for (int j=0; j<5; j++){
                internals[j] = sc.nextInt();
                seeMarks[i].setSEEMarks(sc);
            }
```

```
System.out.println("\n Final Marks of Students");

for (int i=0 ; i<n ; i++){
System.out.println("\n Student "+(i+1)+"\n
+ students[i].name+" (USN : "+students[i].
usn+" )");          Courses || Internal || Sem || Final
System.out.println("
for (int j=0 ; j<5 ; j++){
        int finalMarks = internalMarks[i]
getInternalMarks()[j] + seeMarks[i].getSEEMarks
    System.out.println("Courses "+(j+1)+"\t
+ internalMarks[i].getInternalMarks()[j]+"\t
+seeMarks[i].getSEEMarks()[j]+"\t"+finalMarks
}
}
sc.close();
}
}
```

CIE/

Internals.java

```
package CIE;

public class Internals extends Students{
    public int[] internalMarks = new int[5];
    public void setInternalMarks(int[] marks){
        for(int i=0 ; i<5 ; i++){
            internalMarks[i] = marks[i];
        }
    }
    public int[] getInternalMarks(){
        return internalMarks;
    }
}
```

Students.java

```
package CIE;
public class Student {
    public String usn;
    public String name;
    public int sem;
}
```

Enter USN for Student 2 : 1BM23CS074

Enter Name for Student 2 : Chetan Kumar

Enter Semester for Students : 1

Enter Internal Marks (5 courses) for Student 2

40
40
39
37
32

Enter SEE Marks (5 courses) for Student 2

95
86
81
91
93

Final Marks of Students:

Student 1 : Chetan KS (USN : 1BM23CS074)

Courses	Internal	SEE	Final Marks
Course 1:	40	98	138
Course 2:	36	86	122
Course 3:	39	91	130
Course 4	35	87	122
Courses 5:	29	78	107

Student 2: Chetan Kumar

Course	Internal	SEE	Final Marks
Course 1:	40	95	135
Course 2:	40	86	126
Course 3:	39	81	120
Course 4:	37	91	128
Course 5:	32	93	125

SEE

External.Java

```
package SEE;
import CIE.Student;
public class External extends Student{
public int[] seeMarks = new int[5];
Public void setSEEMarks(int[] marks){
    for(int i=0; i<5; i++){
        seeMarks[i] = marks[i];
    }
}
public int[] getSEEmarks(){
    return seeMarks;
}
}
```

Output

Enter the no of students : 2

Enter USN for student 1: 1BM23CS074

Enter name for Student 1 : Chetan KS

Enter semester for student 1: 1

Enter Internal Marks (5 courses) for Student 1

40
36
39
35
29

Enter SEE Marks (5 courses) for student 1:

98
86
91
87
78

Code:

```
package CIE;

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

package CIE;

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

    public void setInternalMarks(int[] marks) {
        for (int i = 0; i < 5; i++) {
            internalMarks[i] = marks[i];
        }
    }

    public int[] getInternalMarks() {
        return internalMarks;
    }
}

package SEE;

import CIE.Student;

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

    public void setSEEMarks(int[] marks) {
        for (int i = 0; i < 5; i++) {
            seeMarks[i] = marks[i];
        }
    }

    public int[] getSEEMarks() {
        return seeMarks;
```

```
    }
}

import CIE.*;
import SEE.*;

import java.util.Scanner;

public class FinalMarksCalculator {
    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[] internalMarks = new Internals[n];
        External[] seeMarks = new External[n];

        for (int i = 0; i < n; i++) {
            students[i] = new Student();
            internalMarks[i] = new Internals();
            seeMarks[i] = new External();

            System.out.print("Enter USN for Student " + (i + 1) + ": ");
            students[i].usn = sc.next();
            sc.nextLine();

            System.out.print("Enter Name for Student " + (i + 1) + ": ");
            students[i].name = sc.nextLine();

            System.out.print("Enter Semester for Student " + (i + 1) + ": ");
            students[i].sem = sc.nextInt();

            int[] internals = new int[5];
            System.out.println("Enter Internal Marks (5 courses) for Student " + (i + 1) + ": ");
            for (int j = 0; j < 5; j++) {
                internals[j] = sc.nextInt();
            }
            internalMarks[i].setInternalMarks(internals);

            int[] see = new int[5];
            System.out.println("Enter SEE Marks (5 courses) for Student " + (i + 1) + ": ");
            for (int j = 0; j < 5; j++) {
                see[j] = sc.nextInt();
            }
            seeMarks[i].setSEEMarks(see);
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
```

```
C:\Users\Dell\OneDrive\Desktop\StudentMarksProject>javac CIE\Student.java

C:\Users\Dell\OneDrive\Desktop\StudentMarksProject>javac CIE\Internals.java

C:\Users\Dell\OneDrive\Desktop\StudentMarksProject>javac SEE\External.java

C:\Users\Dell\OneDrive\Desktop\StudentMarksProject>javac FinalMarksCalculator.java

C:\Users\Dell\OneDrive\Desktop\StudentMarksProject>java FinalMarksCalculator
Enter the number of students: 2
Enter USN for Student 1: 1BM23CS074
Enter Name for Student 1: CHETHAN K S
Enter Semester for Student 1: 1
Enter Internal Marks (5 courses) for Student 1:
40
36
39
35
29
Enter SEE Marks (5 courses) for Student 1:
98
86
91
87
78
Enter USN for Student 2: 1BM23CS082
Enter Name for Student 2: CREVAN NEIL FERNANDES
Enter Semester for Student 2: 1
Enter Internal Marks (5 courses) for Student 2:
40
40
39
37
32
Enter SEE Marks (5 courses) for Student 2:
95
86
81
91
93

Final Marks of Students:

Student 1: CHETHAN K S (USN: 1BM23CS074)
Course  Internal        SEE     Final Marks
Course 1:       40              98      138
Course 2:       36              86      122
Course 3:       39              91      130
Course 4:       35              87      122
Course 5:       29              78      107
```

System.out.println("\nStudent " + (i + 1) + ": " + students[i].name + " (USN: " + students[i].usn + ")");

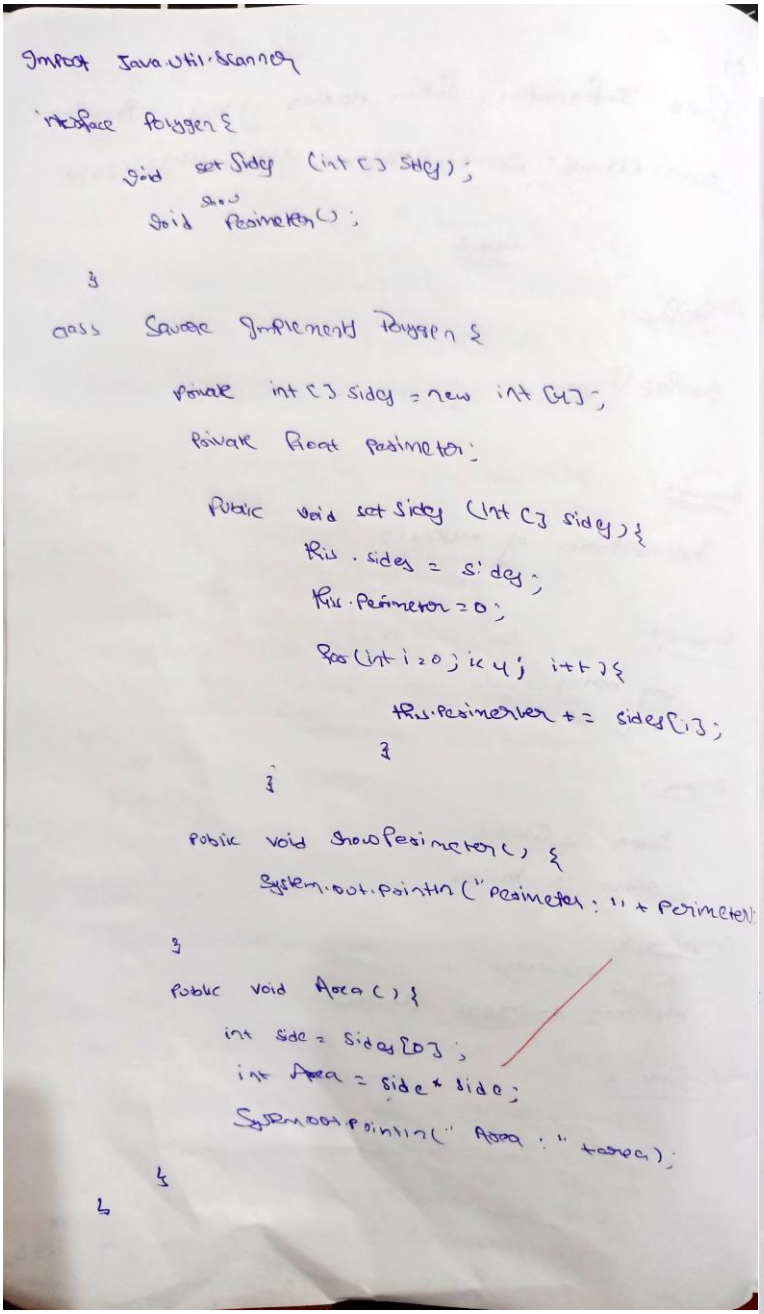
```
System.out.println("Course\tInternal\tSEE\tFinal Marks");
            for (int j = 0; j < 5; j++) {
                int finalMark = internalMarks[i].getInternalMarks()[j] + seeMarks[i].getSEEMarks()[j];
                System.out.println("Course " + (j + 1) + ":\t" + internalMarks[i].getInternalMarks()[j] +
"\t\t" + seeMarks[i].getSEEMarks()[j] + "\t" + finalMark);
            }
        }
            sc.close();
    }
}
```

Output:

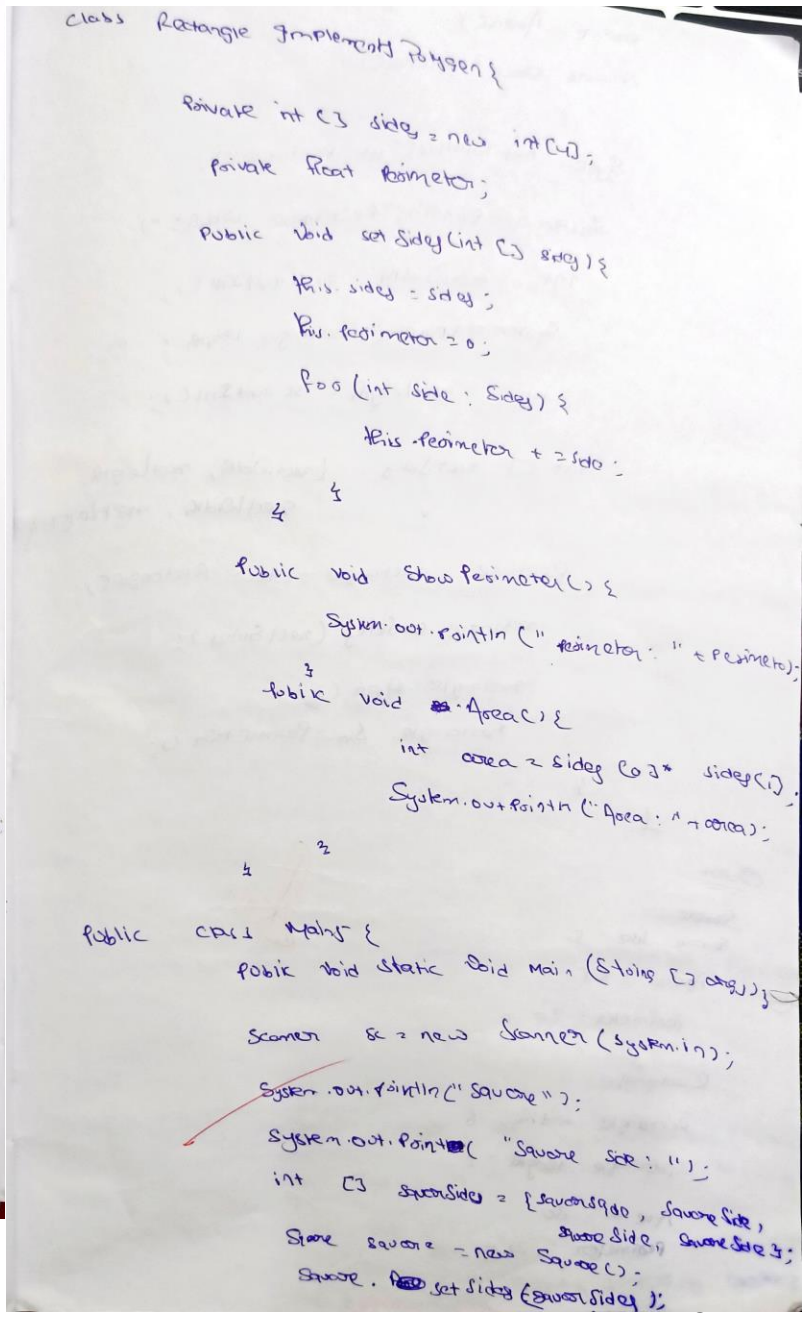
Program 7

Interfaces

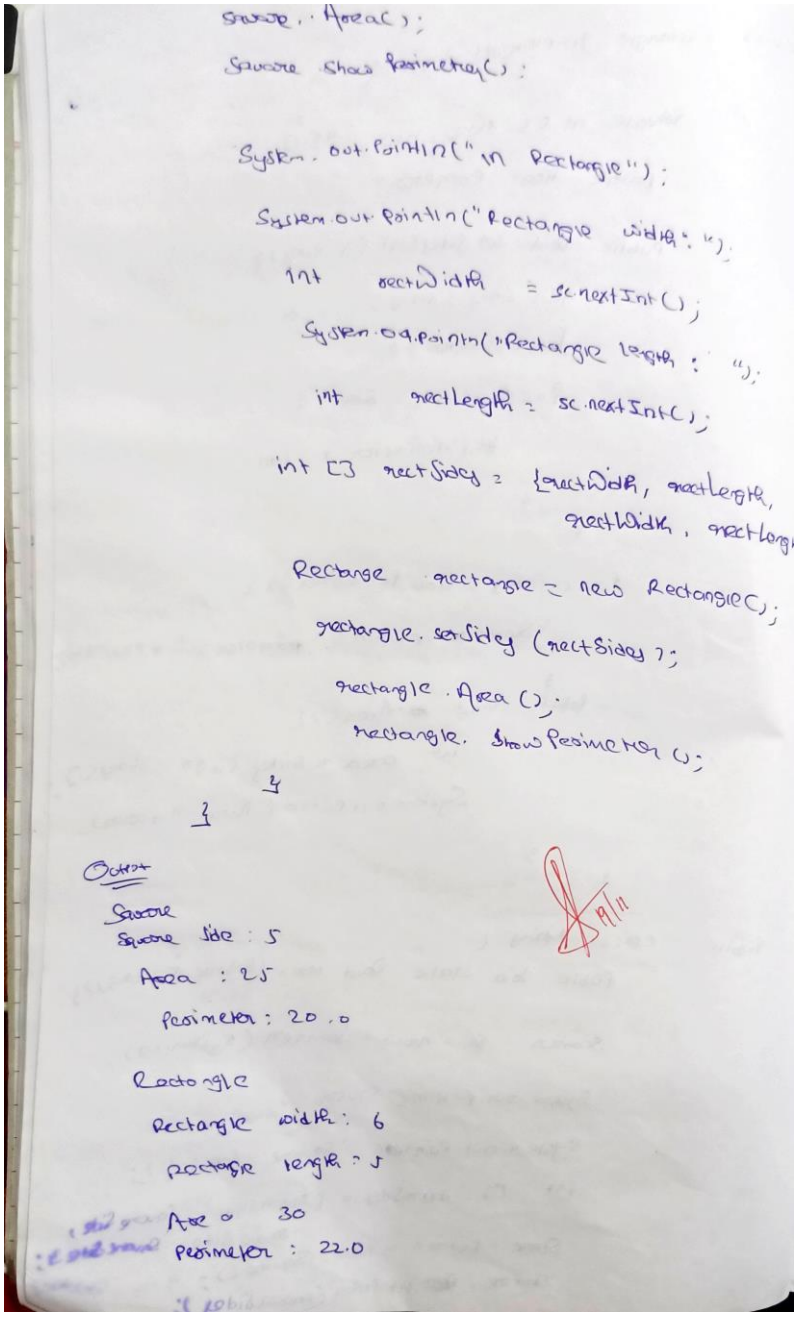
Algorithm:



```
Import java.util.Scanner;
interface Polygon {
    void setSides(int[] sides);
    void showPerimeter();
}
class Square implements Polygon {
    private int[] sides = new int[4];
    private float perimeter;
    public void setSides(int[] sides) {
        this.sides = sides;
        this.perimeter = 0;
        for(int i=0; i<4; i++) {
            this.perimeter += sides[i];
        }
    }
    public void showPerimeter() {
        System.out.println("Perimeter: " + perimeter);
    }
    public void Area() {
        int side = sides[0];
        int area = side * side;
        System.out.println("Area: " + area);
    }
}
```



```
class Rectangle implements Polygon {
    private int[] sides = new int[4];
    private float perimeter;
    public void setSides(int[] sides) {
        this.sides = sides;
        this.perimeter = 0;
        for(int side : sides) {
            this.perimeter += side;
        }
    }
    public void showPerimeter() {
        System.out.println("Perimeter: " + perimeter);
    }
    public void Area() {
        int area = sides[0] * sides[1];
        System.out.println("Area: " + area);
    }
}
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Square");
        System.out.println("Square Side:");
        int[] squareSides = {squareSide, squareSide, squareSide, squareSide};
        Square square = new Square();
        square.setSides(squareSides);
```



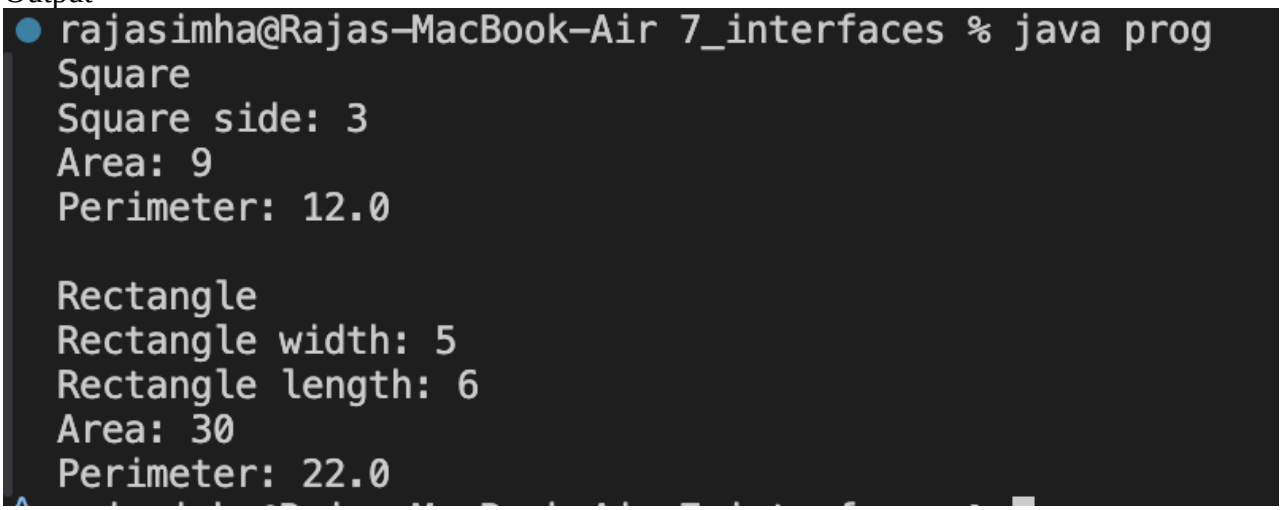
Code:

```
import java.util.Scanner;

interface Polygon {
  void setSides(int[] sides);
```

```
 void showPerimeter();
}

class Square implements Polygon {
    private int[] sides = new int[4];
    private float perimeter;

    public void setSides(int[] sides) {
        this.sides = sides;
        this.perimeter = 0;
        for (int side : sides) {
            this.perimeter += side;
        }
    }

    public void showPerimeter() {
        System.out.println("Perimeter: " + perimeter);
    }

    public void Area() {
        int side = sides[0];
        int area = side * side;
        System.out.println("Area: " + area);
    }
}

class Rectangle implements Polygon {
    private int[] sides = new int[4];
    private float perimeter;

    public void setSides(int[] sides) {
        this.sides = sides;
        this.perimeter = 0;
        for (int side : sides) {
            this.perimeter += side;
        }
    }

    public void showPerimeter() {
        System.out.println("Perimeter: " + perimeter);
    }

    public void Area() {
        int area = sides[0] * sides[1];
        System.out.println("Area: " + area);
    }
}

public class prog {
```

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

        System.out.println("Square");
        System.out.print("Square side: ");
        int squareSide = sc.nextInt();
        int[] squareSides = { squareSide, squareSide, squareSide, squareSide };
        Square square = new Square();
        square.setSides(squareSides);
        square.Area();
        square.showPerimeter();

        System.out.println("\nRectangle");
        System.out.print("Rectangle width: ");
        int rectWidth = sc.nextInt();
        System.out.print("Rectangle length: ");
        int rectLength = sc.nextInt();
        int[] rectSides = { rectWidth, rectLength, rectWidth, rectLength };
        Rectangle rectangle = new Rectangle();
        rectangle.setSides(rectSides);
        rectangle.Area();
        rectangle.showPerimeter();
    }
```

Output

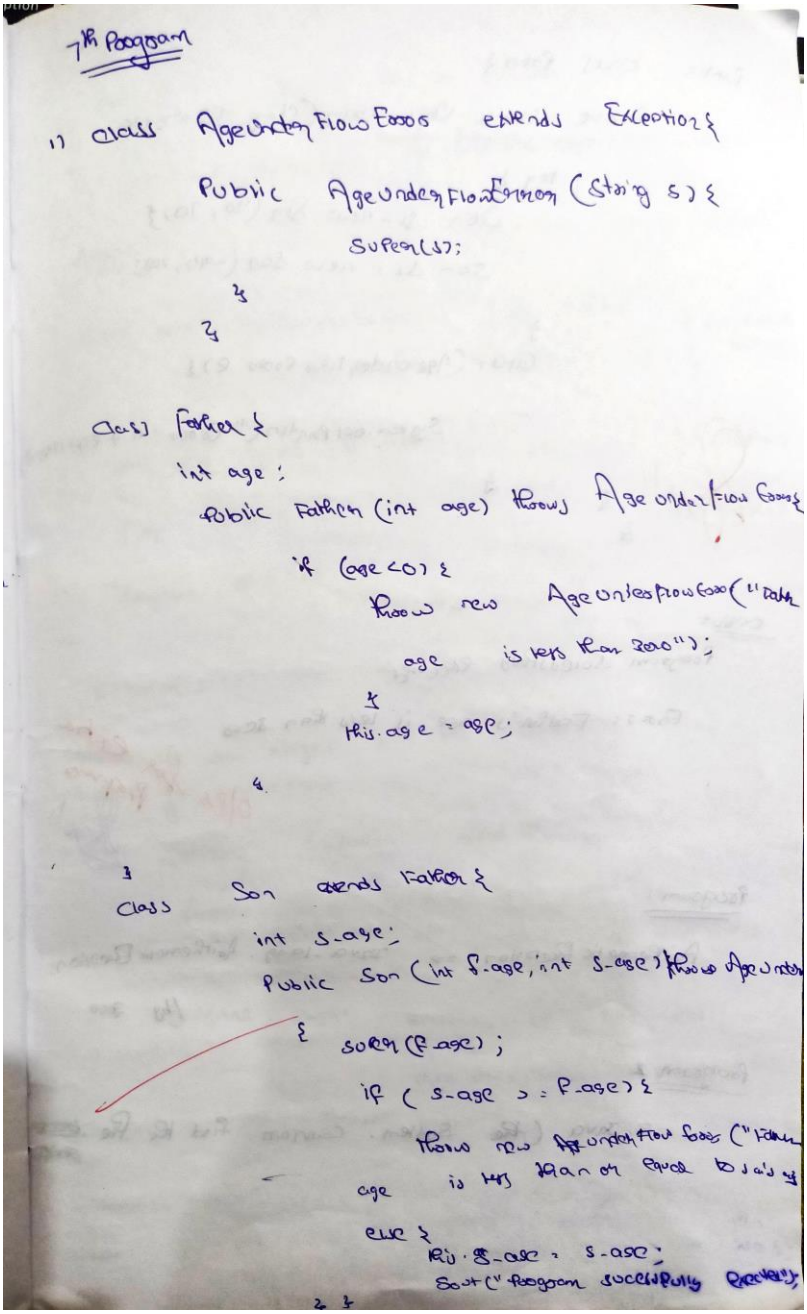


Program 8

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:

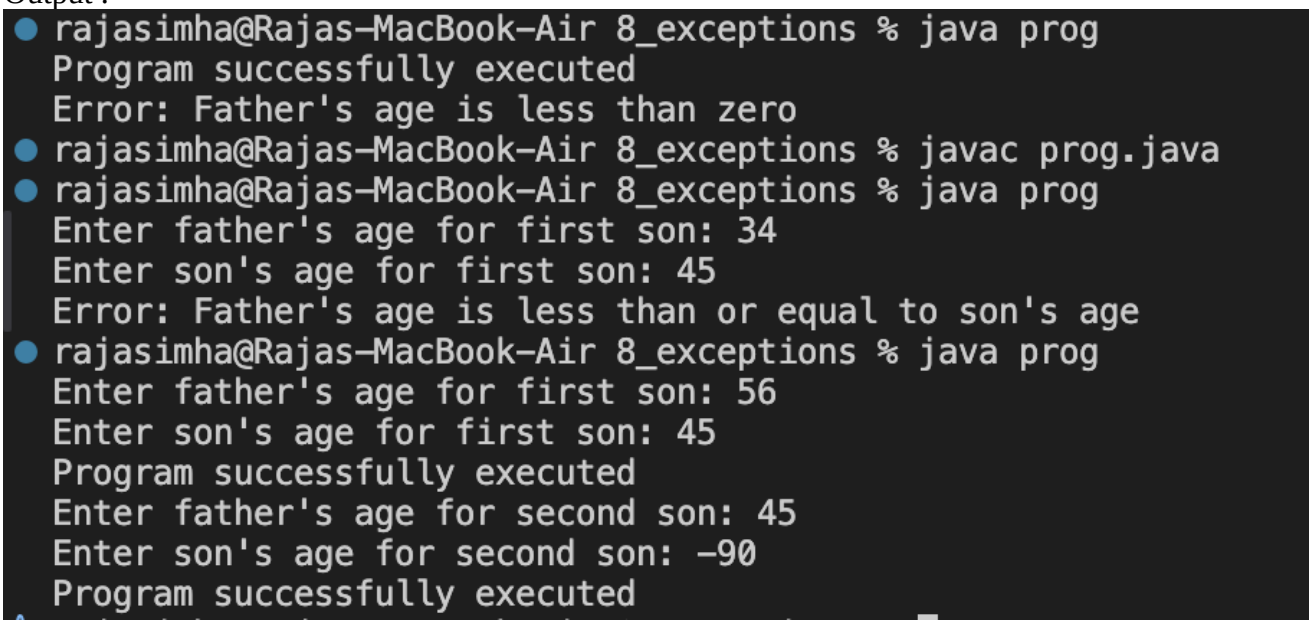

7th Program
class AgeUnderFlowExcep extends Exception{
Pubic AgeUnderFlowExcep (String s){
super(s);
}
}
class Father{
int age;
public Father (int age) throws AgeUnderFlowExcep{
if (age<0){
throw new AgeUnderFlowExcep("Father
age is less than zero");
}
this.age = age;
}
class Son extends Father{
int s_age;
Public Son (int f_age, int s_age) throws AgeUnder
{ super(f_age);
if (s_age >= f_age){
throw new AgeUnderFlowExcep ("Father
age is less than or equal to son's
else {
this.s_age = s_age;
Sout("Program successfully Executed");

Code:

```
class AgeUnderFlowError extends Exception {
   public AgeUnderFlowError(String s) {
      super(s);
   }
}

class Father {
   int age;

   public Father(int age) throws AgeUnderFlowError {
      if (age < 0) {
         throw new AgeUnderFlowError("Father's age is less than zero");
      }
      this.age = age;
   }
}

class Son extends Father {
   int s_age;

   public Son(int f_age, int s_age) throws AgeUnderFlowError {
      super(f_age);

      if (s_age >= f_age) {
         throw new AgeUnderFlowError("Father's age is less than or equal to son's age");
      } else {
         this.s_age = s_age;
         System.out.println("Program successfully executed");
      }
   }
}

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

         System.out.print("Enter father's age for first son: ");
         int f_age1 = scanner.nextInt();
         System.out.print("Enter son's age for first son: ");
         int s_age1 = scanner.nextInt();
         Son s1 = new Son(f_age1, s_age1);

         System.out.print("Enter father's age for second son: ");
         int f_age2 = scanner.nextInt();
         System.out.print("Enter son's age for second son: ");
         int s_age2 = scanner.nextInt();
```

```
            Son s2 = new Son(f_age2, s_age2);

            scanner.close();
        } catch (AgeUnderFlowError e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Output :

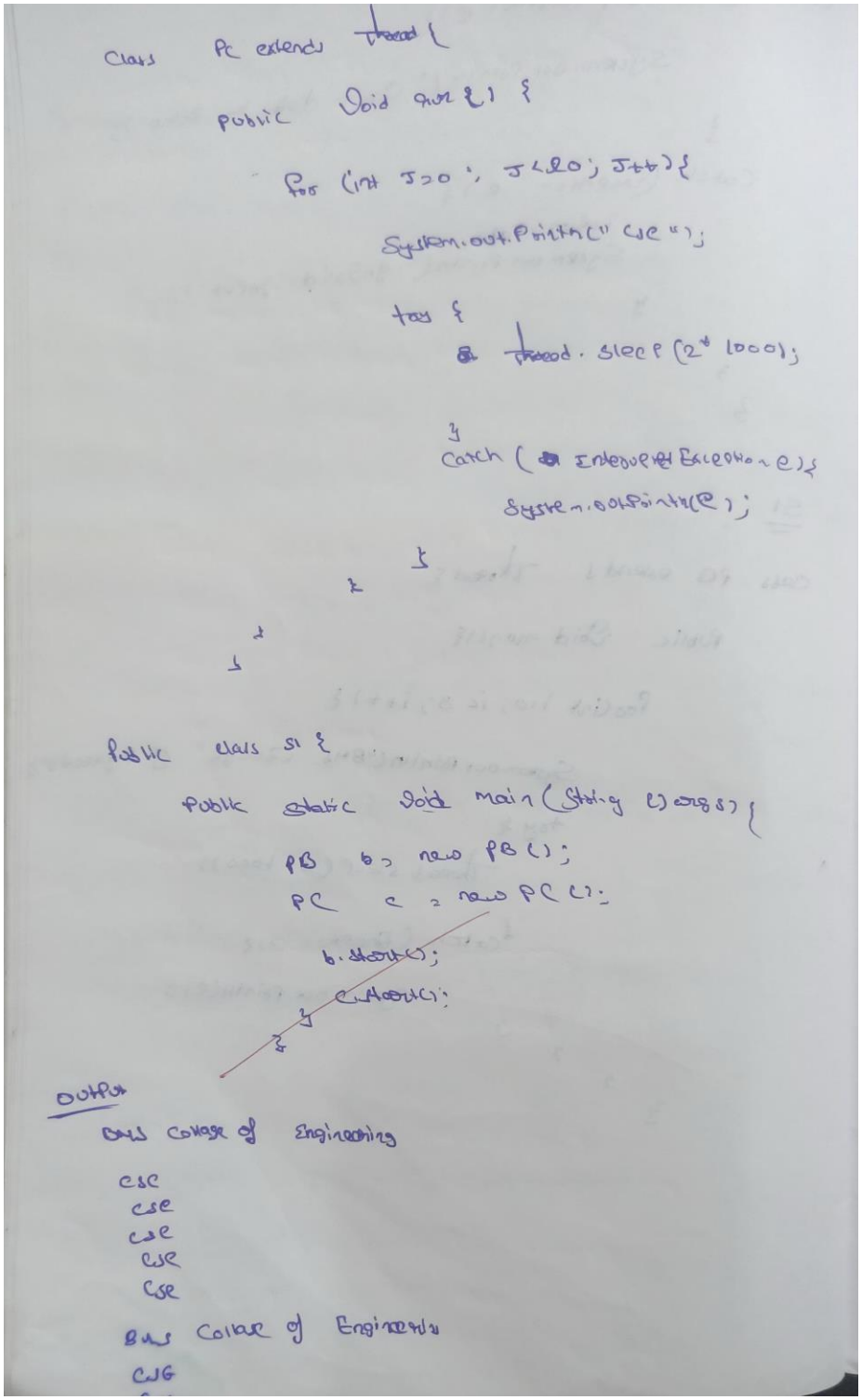


Program 9

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.

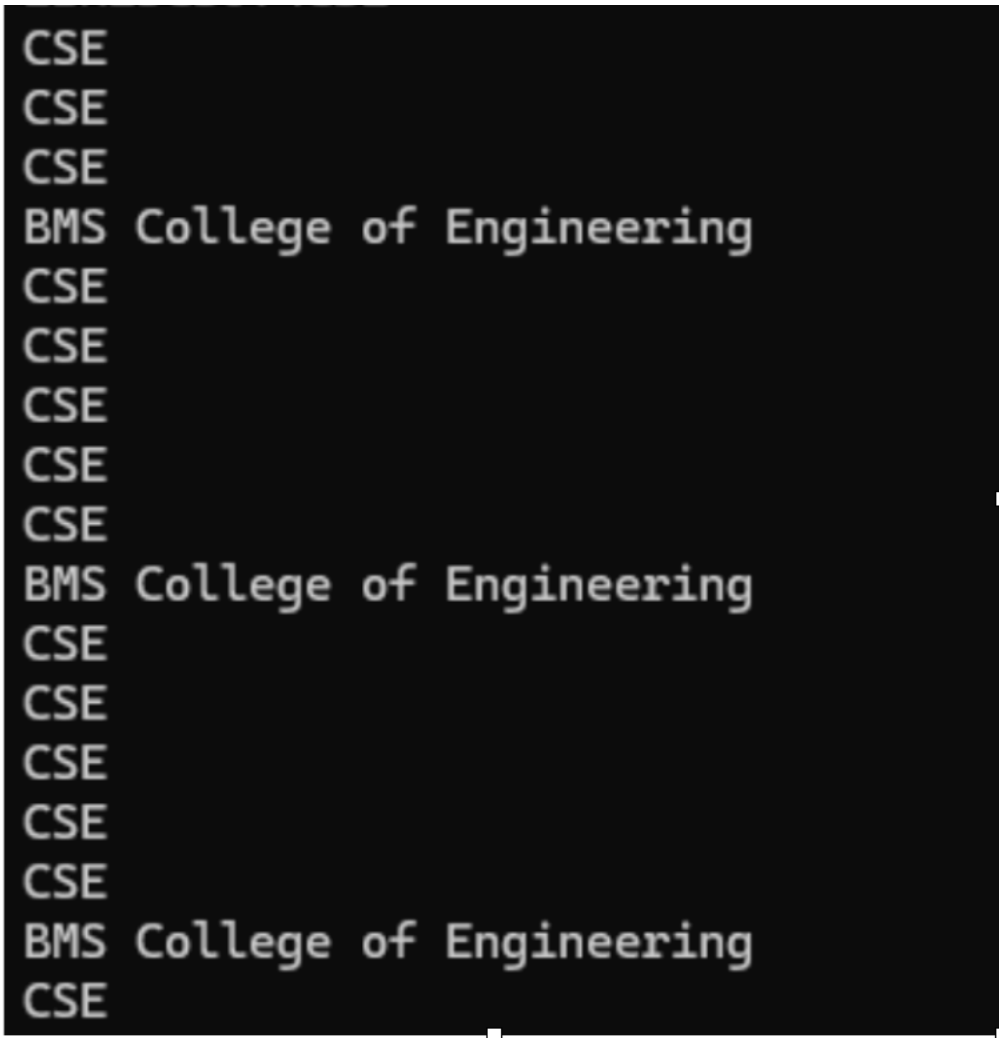
Algorithm:


class PC extends Thread {
public void run() {
for (int j=0; j<10; j++){
System.out.println("CSE");
try {
Thread.sleep(2*1000);
}
catch (InterruptedException e){
System.out.println(e);
}
}
}
public class S1 {
public static void main(String[] args) {
PB b = new PB();
PC c = new PC();
b.start();
c.start();
}
}
Output
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE

Code:

```
class Message1 extends Thread {
   public void run() {
      while (true) {
         try {
            Thread.sleep(10000);
         } catch (InterruptedException e) {
            System.out.println(e);
         }
         System.out.println("BMS College of Engineering");
      }
   }
}

class Message2 extends Thread {
   public void run() {
      while (true) {
         try {
            Thread.sleep(2000);
         } catch (InterruptedException e) {
            System.out.println(e);
         }
         System.out.println("CSE");
      }
   }
}

public class labthread {
   public static void main(String[] args) {
      Message1 thread1 = new Message1();
      Message2 thread2 = new Message2();

      thread1.start();
      thread2.start();
   }
}
```

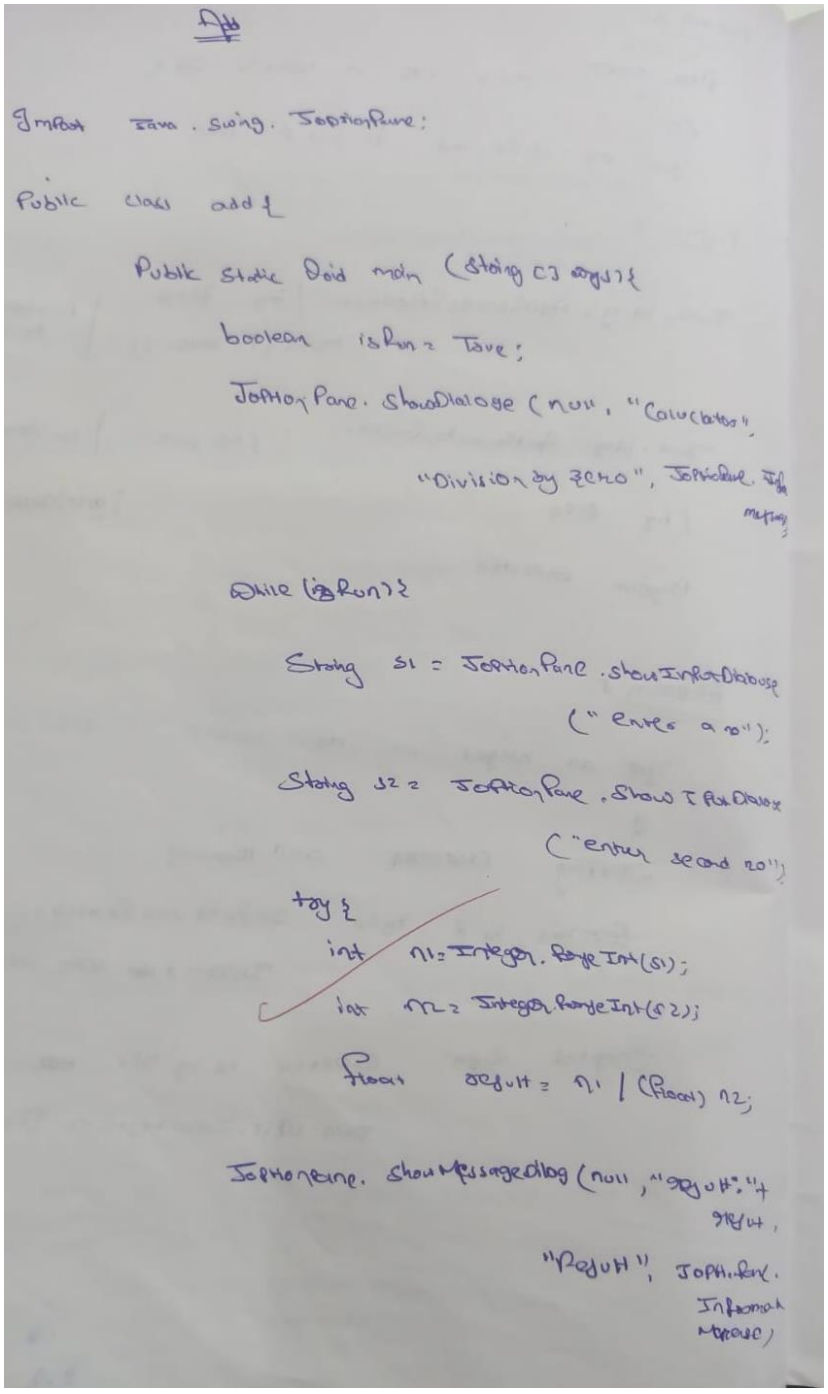
Output:



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.JOptionPane;

public class add {

    public static void main (String[] args){

        boolean isRun = true;

        JOptionPane.showDialog(null, "Calculator",
                "Division by zero", JOptionPane.INFORMATION_MESSAGE);

        while (isRun){

            String s1 = JOptionPane.showInputDialog
                    ("enter a no");

            String s2 = JOptionPane.showInputDialog
                    ("enter second no");

            try {
                int n1 = Integer.parseInt(s1);
                int n2 = Integer.parseInt(s2);

                float result = n1 / (float) n2;

            JOptionPane.showMessageDialog(null, "result:"+
                    result,
                    "Result", JOptionPane.
                    INFORMATION_MESSAGE);
```

Code:

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

public class DivisionApp {
   public static void main(String[] args) {
      JFrame frame = new JFrame("Division Calculator");
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setSize(400, 200);
      frame.setLayout(new GridLayout(4, 2, 10, 10));
      JLabel labelNum1 = new JLabel("Enter number 1:");
      JTextField textNum1 = new JTextField();
      JLabel labelNum2 = new JLabel("Enter number 2:");
      JTextField textNum2 = new JTextField();
      JLabel labelResult = new JLabel("Result:");
      JTextField textResult = new JTextField();
      textResult.setEditable(false);

      JButton buttonDivide = new JButton("Divide");
      frame.add(labelNum1);
      frame.add(textNum1);
      frame.add(labelNum2);
      frame.add(textNum2);
      frame.add(labelResult);
      frame.add(textResult);
      frame.add(new JLabel());
      frame.add(buttonDivide);
      buttonDivide.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
            try {
               int num1 = Integer.parseInt(textNum1.getText());
               int num2 = Integer.parseInt(textNum2.getText());
               if (num2 == 0) {
                  throw new ArithmeticException("Cannot divide by zero.");
               }
               int result = num1 / num2;
               textResult.setText(String.valueOf(result));
            } catch (NumberFormatException ex) {
               JOptionPane.showMessageDialog(frame,
                     "Invalid input! Please enter integers only.",
                     "Number Format Error",
                     JOptionPane.ERROR_MESSAGE);
            } catch (ArithmeticException ex) {
               JOptionPane.showMessageDialog(frame,
```

```
                    ex.getMessage(),
                    "Arithmetic Error",
                    JOptionPane.ERROR_MESSAGE);
            }
        }
    });
    frame.setVisible(true);
  }
}
```

Output: