VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

JAWIN ROYS FERNANDES (1BM23CS122)

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 **JAWIN ROYS FERNANDES (1BM23CS122),** 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.

Dr. Seema Patil Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

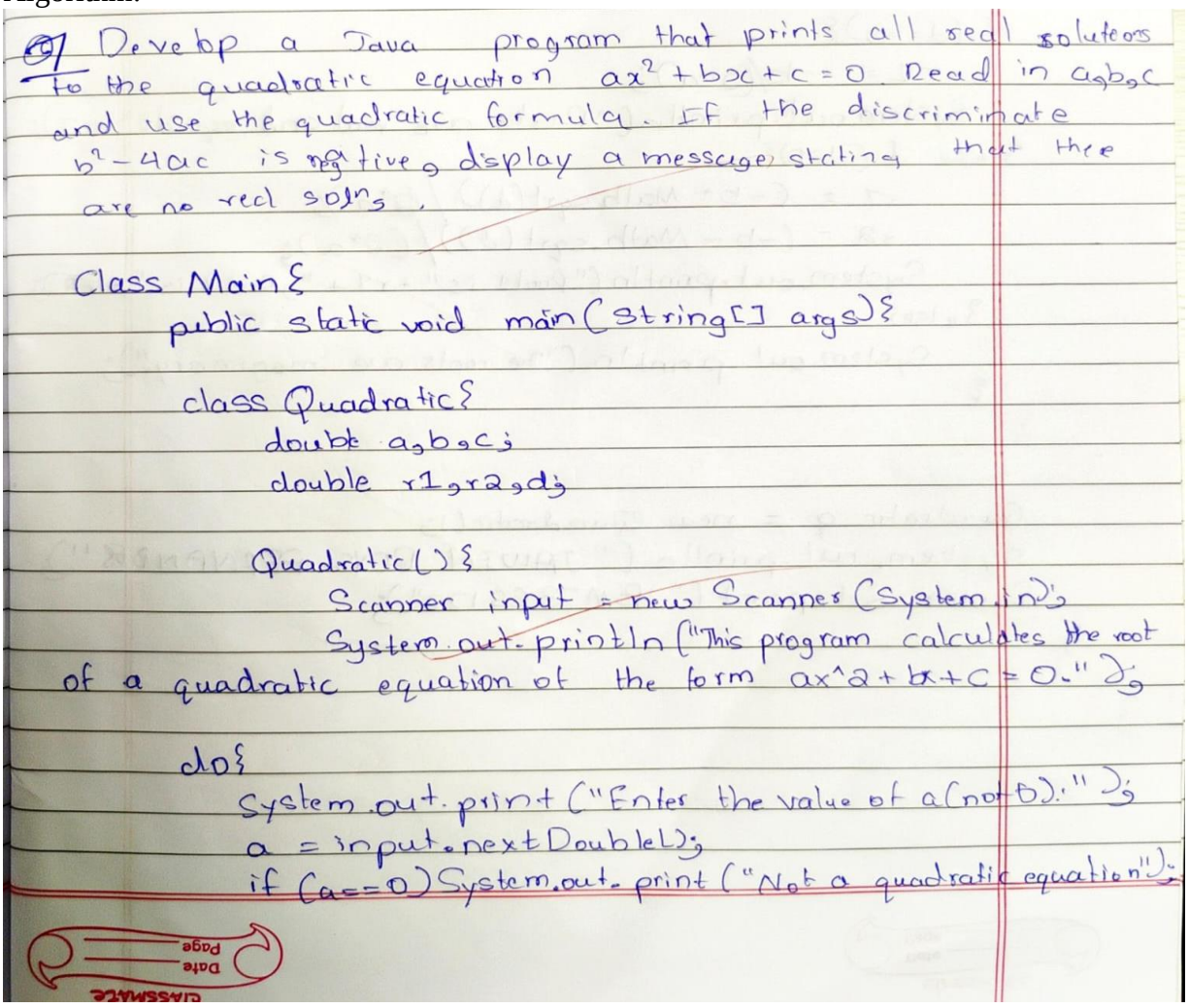
Index

Github Link:
https://github.com/jawin00/Lab-Programs

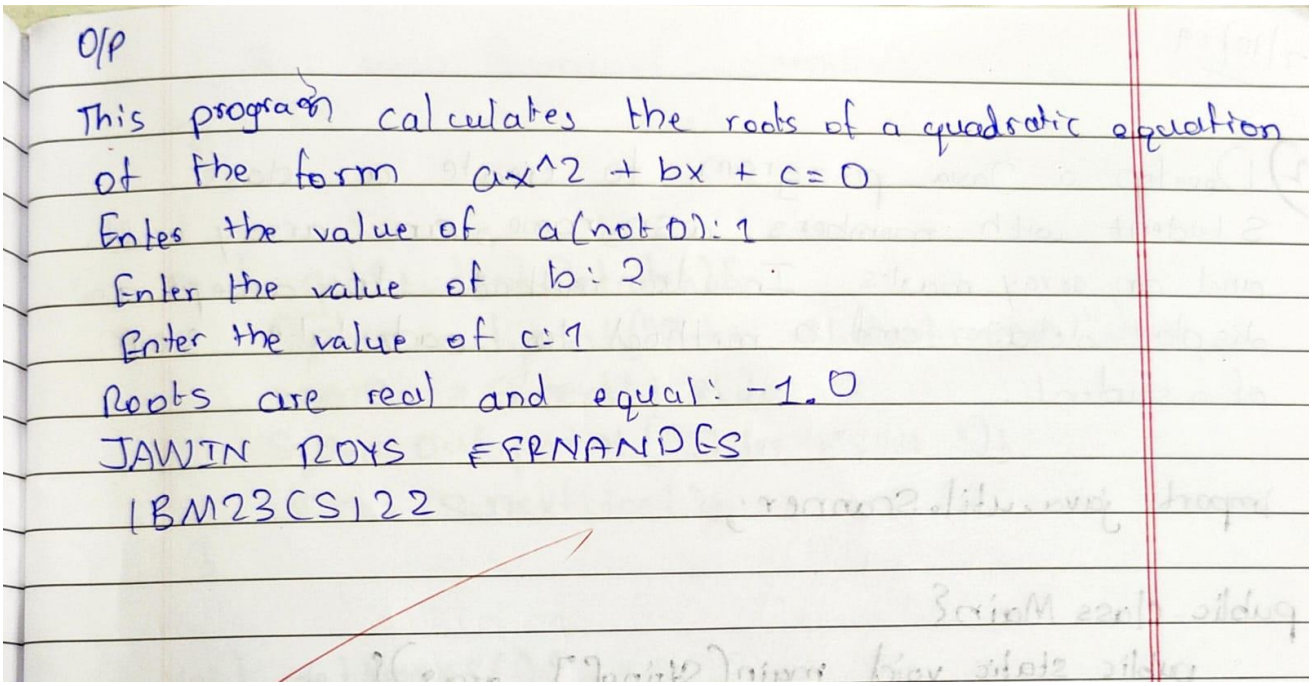
Program 1

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.

Algorithm:



```
} while (a==0);

System.out.print("Enter the value of b: ");
b = input.nextDouble();

System.out.print("Enter the value of b: ");
b= input.nextDouble();

System.out.print("Enter the value of c: ");
c = input.nextDouble();

d = b*b-4*a*c;

if (d==0){
    r1= -b/(2*a);
    System.out.println("Roots are real and equal: "+r1
} else if (d>0){
    r1 = (-b+Math.sqrt(d))/(2*a);
    r2 = (-b-Math.sqrt(d))/(2*a);
    System.out.println("Root1 is "+r1+", Root2 is "+r2
}else{
    System.out.println("The roots are imaginary.");
}
}
}

Quadratic q = new Quadratic();
System.out.println("JAWIN ROYS FERNANDES");
System.out.print("1BM23CS122");
}

}
```



Code:

```
import java.util.Scanner;

class Main {
   public static void main(String[] args) {
class Quadratic {
   double a, b, c;
   double r1, r2, d;

   Quadratic() {
      Scanner input = new Scanner(System.in);
      System.out.println("This program calculates the roots of a quadratic equation of the form ax^2 +
bx + c = 0.");

      do {
         System.out.print("Enter the value of a (not 0): ");
         a = input.nextDouble();
            if(a==0) System.out.print("Not a quadratic equation" )  ;
      } while (a == 0);

      System.out.print("Enter the value of b: ");
      b = input.nextDouble();

      System.out.print("Enter the value of c: ");
      c = input.nextDouble();

      d = b*b-4*a*c;
```

```
        if(d==0) {
            r1 = -b/(2*a);
            System.out.println("Roots are real and equal: " + r1);
        }else if(d > 0){
            r1 = (-b+Math.sqrt(d))/(2*a);
            r2 = (-b-Math.sqrt(d))/(2 * a);
            System.out.println("Root1 is " + r1 + ", Root2 is " + r2);
        } else {
            System.out.println("The roots are imaginary.");
        }
    }
}
        Quadratic q = new Quadratic();
            System.out.println("JAWIN ROYS FERNANDES" )  ;
            System.out.print("1BM23CS122") ;
    }
}
```

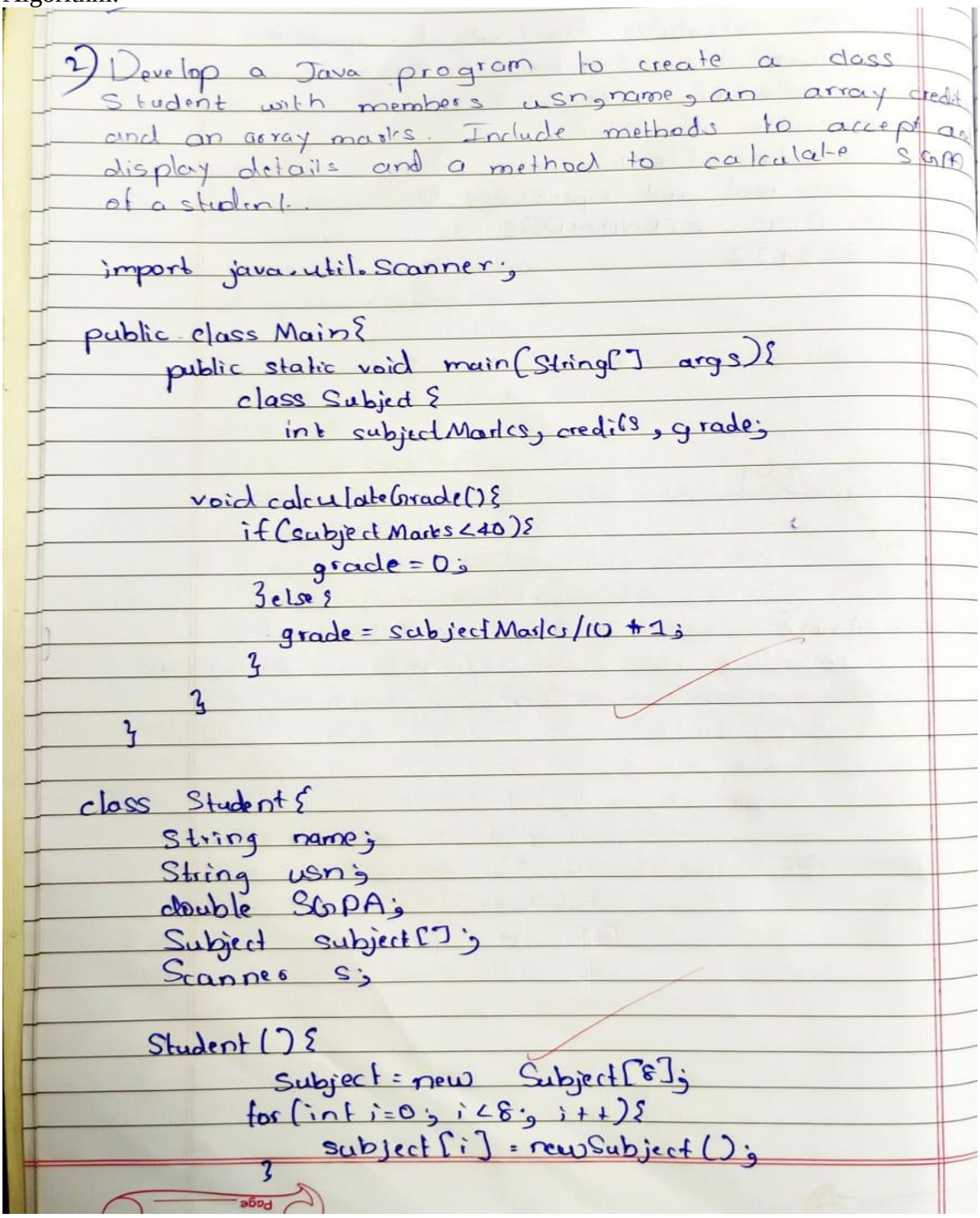
```
PS D:\1BM23CS122\quadratic> javac Quadratic.java
PS D:\1BM23CS122\quadratic> java Quadratic.java
This program calculates the roots of a quadratic equation of the form ax^2 + bx + c = 0.
Enter the value of a (not 0): 1
Enter the value of b: 3
Enter the value of c: 2
Root1 is -1.0, Root2 is -2.0
JAWIN ROYS FERNANDES
1BM23CS122
PS D:\1BM23CS122\quadratic> java Quadratic.java
This program calculates the roots of a quadratic equation of the form ax^2 + bx + c = 0.
Enter the value of a (not 0): 1
Enter the value of b: 2
Enter the value of c: 1
Roots are real and equal: -1.0
JAWIN ROYS FERNANDES
1BM23CS122
```

```
PS D:\1BM23CS122\quadratic> java Quadratic.java
This program calculates the roots of a quadratic equation of the form ax^2 + bx + c = 0.
Enter the value of a (not 0): 5
Enter the value of b: 1
Enter the value of c: 6
The roots are imaginary.
JAWIN ROYS FERNANDES
1BM23CS122
```

Program 2

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

Algorithm:



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

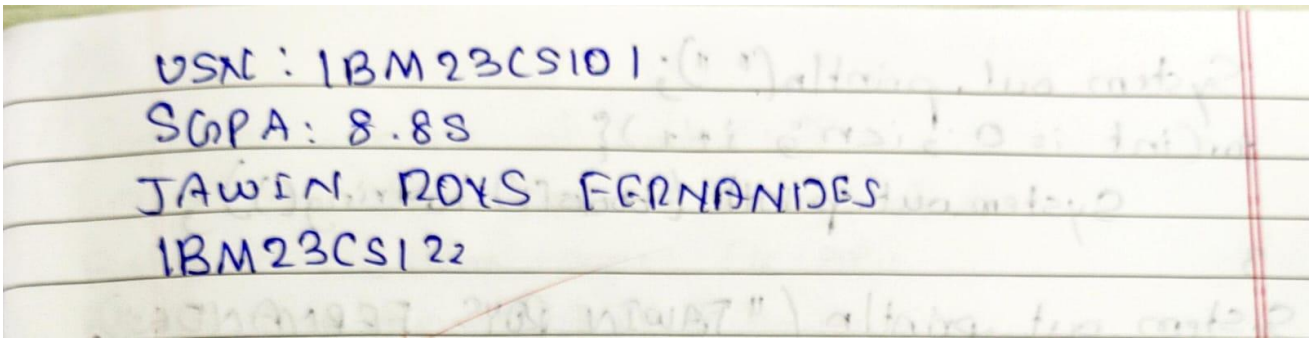
```
import java.util.Scanner;

public class Main{
    public static void main(String[] args){
        class Subject {
            int subjectMarks, credits, grade;

        void calculateGrade(){
            if(subjectMarks<40){
                grade = 0;
            }else{
                grade = subjectMarks/10 +1;
            }
        }
    }

class Student{
    String name;
    String usn;
    double SGPA;
    Subject subject[];
    Scanner s;

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

```
        s = new Scanner(System.in);
    }

    void getStudentDetails(){
        System.out.print("Enter student name: ");
        name = s.nextLine();
        System.out.print("Enter USN: ");
        usn = s.nextLine();
    }

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

            subject[i].calculateGrade();
            if(subject[i].subjectMarks > 100){
                System.out.println("Marks should not exceed 100");
            }
        }
    }

    void computeSGPA(){
        int totalCredits = 0, totalPoints = 0;

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

        SGPA = (double) totalPoints / totalCredits;
    }
```

```
    void displayResult(){
        System.out.println("Student Name: "+name);
        System.out.println("USN: " + usn);
        System.out.println("SGPA: " + SGPA);
    }
}
    Student S1= new Student();
    S1.getStudentDetails();
    S1.getMarks();
    S1.computeSGPA();
    S1.displayResult();
    System.out.println("JAWIN ROYS FERNANDES");
    System.out.print("1BM23CS122");
}
}
```

O/P:

```
Enter student name : Harsh
Enter USN : 1BM23CS101
Enter marks for Subject 1: 90
Enter credits for Subjects 1: 4
Enter marks for Subject 2: 85
Enter credits for subject 2: 3
Enter marks for subject 3: 85
Enter credits for subject 3: 2
Enter marks for subject 4: 69
Enter credits for subject 4: 3
Enter marks for subject 5: 90
Enter credits for subject 5: 4
Enter marks for subject 6: 90
Enter credits for subject 6: 4
Enter marks for subject 7: 56
Enter Credits for subject 7: 3
Enter marks for subject 8: 86
Enter credits for Subject 8: 2
Student Name: Harsh
```



Code:

```
import java.util.Scanner;


public class Main {
    public static void main(String[] args) {
            class Subject {
    int subjectMarks, credits, grade;

    void calculateGrade() {
        if (subjectMarks < 40) {
            grade = 0;
        } else {
            grade = subjectMarks / 10 + 1;
        }
    }
}

class Student {
    String name;
    String usn;
    double SGPA;
    Subject subject[];
    Scanner s;

    Student() {
        subject = new Subject[8];
        for (int i = 0; i < 8; i++) {
            subject[i] = new Subject();
        }
        s = new Scanner(System.in);
    }

    void getStudentDetails() {
        System.out.print("Enter student name: ");
        name = s.nextLine();
        System.out.print("Enter USN: ");
        usn = s.nextLine();
```

```
    }

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

            subject[i].calculateGrade();
            if (subject[i].subjectMarks > 100) {
                System.out.println("Marks should not exceed 100");
            }
        }
    }

    void computeSGPA() {
        int totalCredits = 0, totalPoints = 0;

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

        SGPA = (double)totalPoints / totalCredits;
    }

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

        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        s1.displayResult();
            System.out.println("JAWIN ROYS FERNANDES" )  ;
            System.out.print("1BM23CS122") ;
    }
}
```

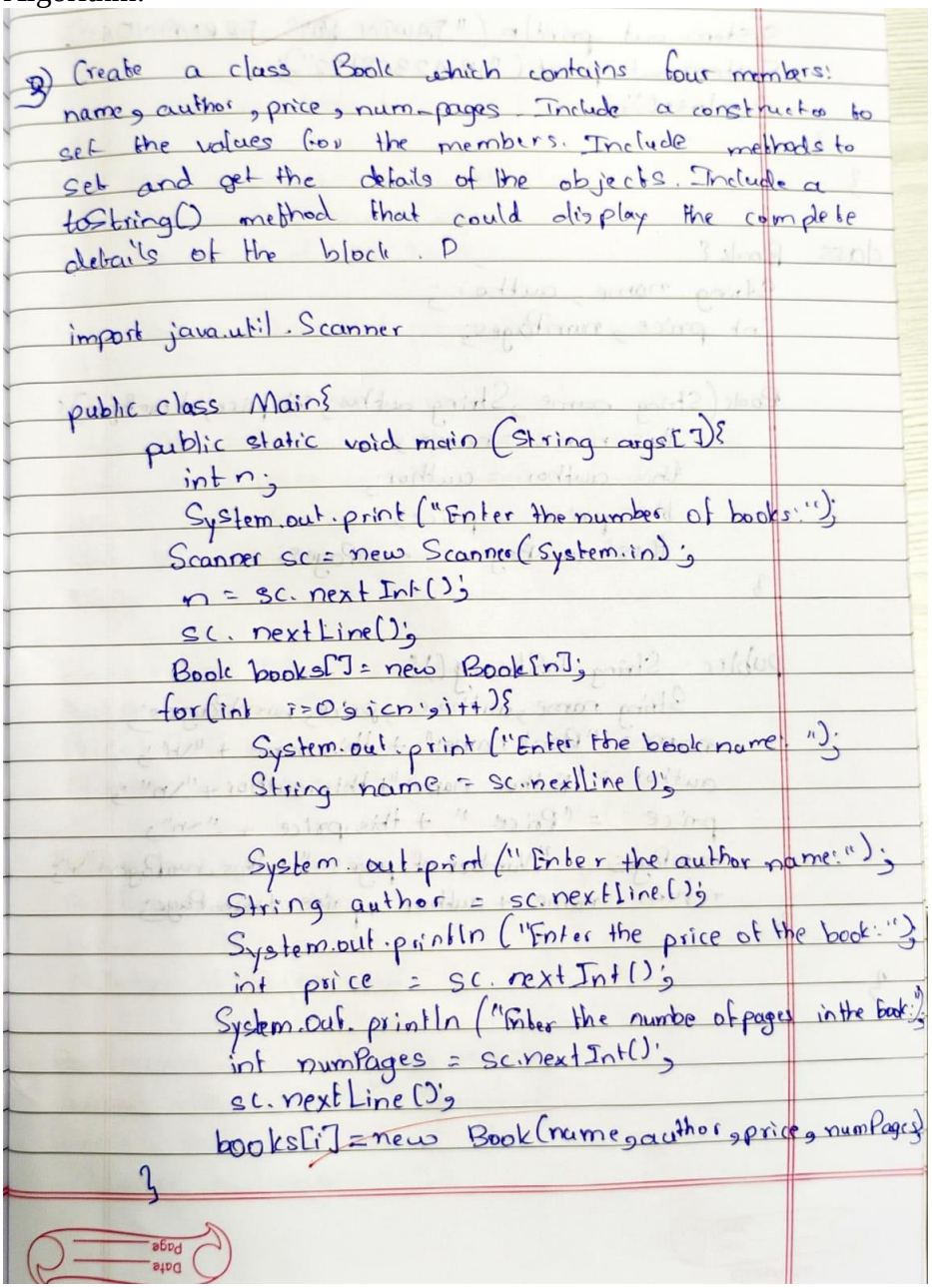
```
Enter student name: Harsh
Enter USN: 1BM23CS101
Enter marks for Subject 1: 90
Enter credits for Subject 1: 4
Enter marks for Subject 2: 85
Enter credits for Subject 2: 3
Enter marks for Subject 3: 85
Enter credits for Subject 3: 2
Enter marks for Subject 4: 69
Enter credits for Subject 4: 3
Enter marks for Subject 5: 90
Enter credits for Subject 5: 4
Enter marks for Subject 6: 90
Enter credits for Subject 6: 4
Enter marks for Subject 7: 56
Enter credits for Subject 7: 3
Enter marks for Subject 8: 86
Enter credits for Subject 8: 2
Student Name: Harsh
USN: 1BM23CS101
SGPA: 8.88
JAWIN ROYS FERNANDES
1BM23CS122
```

```
PS D:\1BM23CS122> java Main.java
Enter student name: CHETHAN
Enter USN: 1BM23CS010
Enter marks for Subject 1: 85
Enter credits for Subject 1: 3
Enter marks for Subject 2: 69
Enter credits for Subject 2: 4
Enter marks for Subject 3: 79
Enter credits for Subject 3: 3
Enter marks for Subject 4: 45
Enter credits for Subject 4: 3
Enter marks for Subject 5: 90
Enter credits for Subject 5: 4
Enter marks for Subject 6: 95
Enter credits for Subject 6: 2
Enter marks for Subject 7: 56
Enter credits for Subject 7: 2
Enter marks for Subject 8: 69
Enter credits for Subject 8: 2
Student Name: CHETHAN
USN: 1BM23CS010
SGPA: 7.826086956521739
JAWIN ROYS FERNANDES
1BM23CS122
```

Program 3

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

Algorithm:



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

```
import java.util.Scanner

public class Main{
    public static void main (String args[]){
        int n;
        System.out.print("Enter the number of books:");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        sc.nextLine();
        Book books[] = new Book[n];
        for(int i=0;i<n;i++){
            System.out.print("Enter the book name:");
            String name = sc.nextline();

            System.out.print("Enter the author name:");
            String author = sc.nextLine();
            System.out.println("Enter the price of the book:");
            int price = sc.nextInt();
            System.out.println("Enter the numbe of pages in the book:");
            int numPages = sc.nextInt();
            sc.nextLine();
            books[i] = new Book(name,author,price,numPages);
        }
```

```
        System.out.println(" ");
        for(int i= 0 ;i<n ; i++){
            System.out.println(books[i].toString());
        }
        System.out.println("JAWIN ROYS FERNANDES");
        System.out.print("1BM23CS122");
        sc.close();
    }
}

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

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

    public String toString(){
        String name, author, price, numPages;
        name = "Book name:" + this.name + "\n";
        author = "Author name:" + this.author + "\n";
        price = "Price " + this.price + "\n";
        numPages = "Number of pages:" + this.numPages + "\n";
        return name + author + price + numPage;
    }
}
```

Q8

```
Enter the number of books : 3
Enter the bookname : Wings of fire
Enter the author name : Dr APJ
Enter the price of the book:
345
Enter the number of pages in the book:
456
Enter the bookname: Atomic Habits
Enter the author name: xyz
Enter the price of the book:
567
Enter the number of pages in the book:
456
Enter the bookname: Puzzling Problems
Enter the author name: Martin Gardener
Enter the price of the book:
456
Enter the number of pages in the book:
122

Book name : Wings of fire
Author name: Dr APJ
Price: 345
Number of pages : 456

Bookname: Atomic Habits
Author name : xyz
Price : 567
Number of pages : 456

Bookname : Puzzling problems
Author name: Martin Gardener
Price : 456
Number of pages : 122
```

Code:

```
import java.util.Scanner  ;


public class Main{
	public static void main(String args[]){
		int n ;
		System.out.print("Enter the number of books:") ;
		Scanner sc  = new Scanner(System.in) ; n = sc.nextInt() ;
		sc.nextLine() ;
		Book books[] = new Book[n];
		for(int i = 0 ; i<n ; i++){
			System.out.print("Enter the book name: ") ;
			String name = sc.nextLine() ;

			System.out.print("Enter the author name:") ;
			String author = sc.nextLine()  ;
			System.out.println("Enter the price of the book:") ;
			int price = sc.nextInt() ;
			System.out.println("Enter the number of pages in the book:") ;
			int numPages = sc.nextInt() ;
			sc.nextLine() ;

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

		System.out.println("");
		 for(int i = 0 ; i<n ; i++){
			System.out.println(books[i].toString()) ;
		}
		System.out.println("JAWIN ROYS FERNANDES" )  ;
		System.out.print("1BM23CS122") ;
		sc.close();
	}
}


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

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

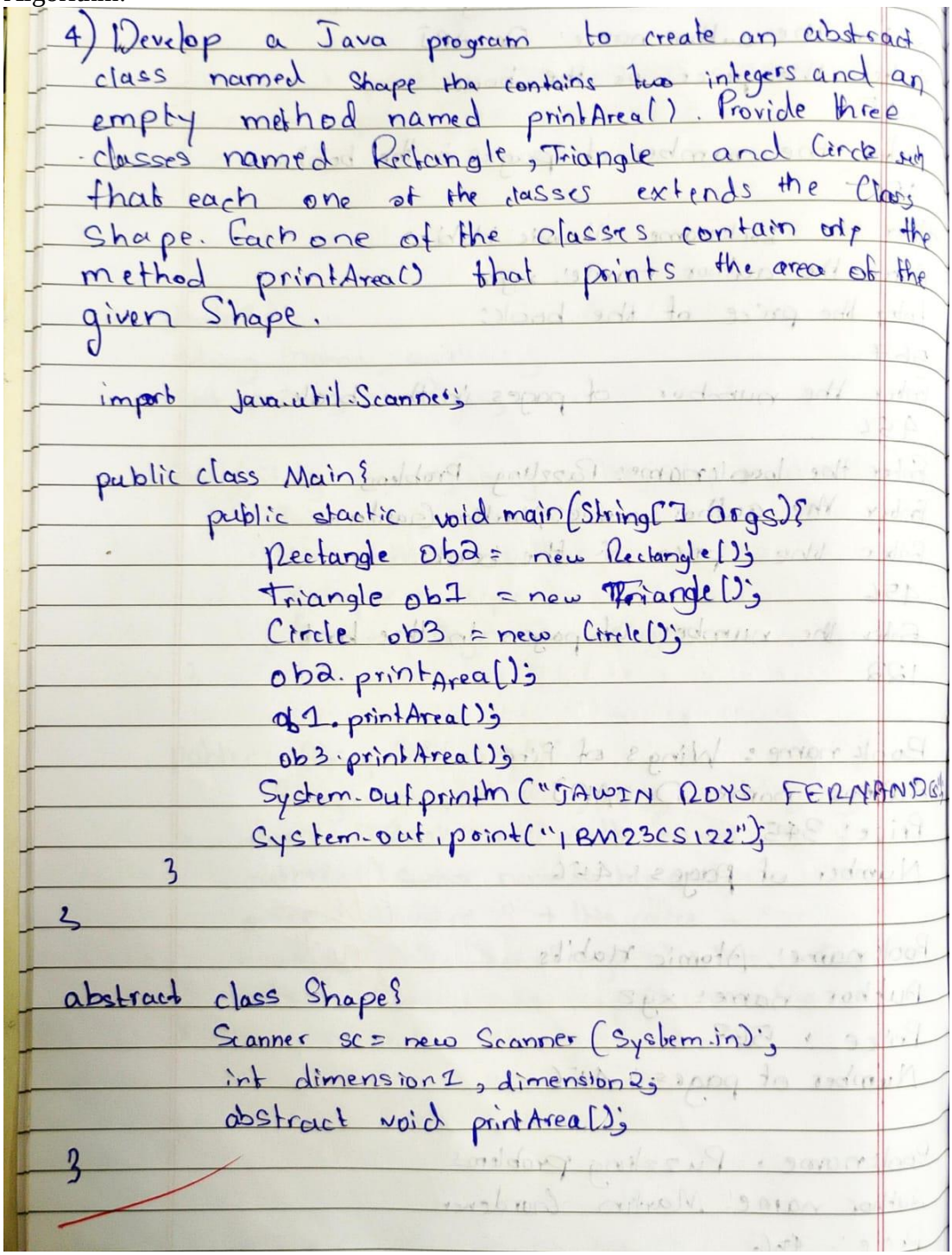
```
        public String toString(){
                String name  ,author  , price,numPages ;
                name = "Book name: " + this.name + "\n" ;
                author = "Author name: " + this.author + "\n" ;
                price = "Price: " + this.price + "\n" ;
                numPages = "Number of pages: " + this.numPages + "\n" ;
                return name + author + price + numPages ;
        }
}
```

```
Enter the number of books:3
Enter the book name: Wings of Fire
Enter the author name:Dr APJ
Enter the price of the book:
345
Enter the number of pages in the book:
456
Enter the book name: Atomic Habits
Enter the author name:xyz
Enter the price of the book:
567
Enter the number of pages in the book:
456
Enter the book name: Puzzling Problems
Enter the author name:Martin Gardener
Enter the price of the book:
456
Enter the number of pages in the book:
122

Book name: Wings of Fire
Author name: Dr APJ
Price: 345
Number of pages: 456

Book name: Atomic Habits
Author name: xyz
Price: 567
Number of pages: 456

Book name: Puzzling Problems
Author name: Martin Gardener
Price: 456
Number of pages: 122

JAWIN ROYS FERNANDES
1BM23CS122
```

Program 4

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

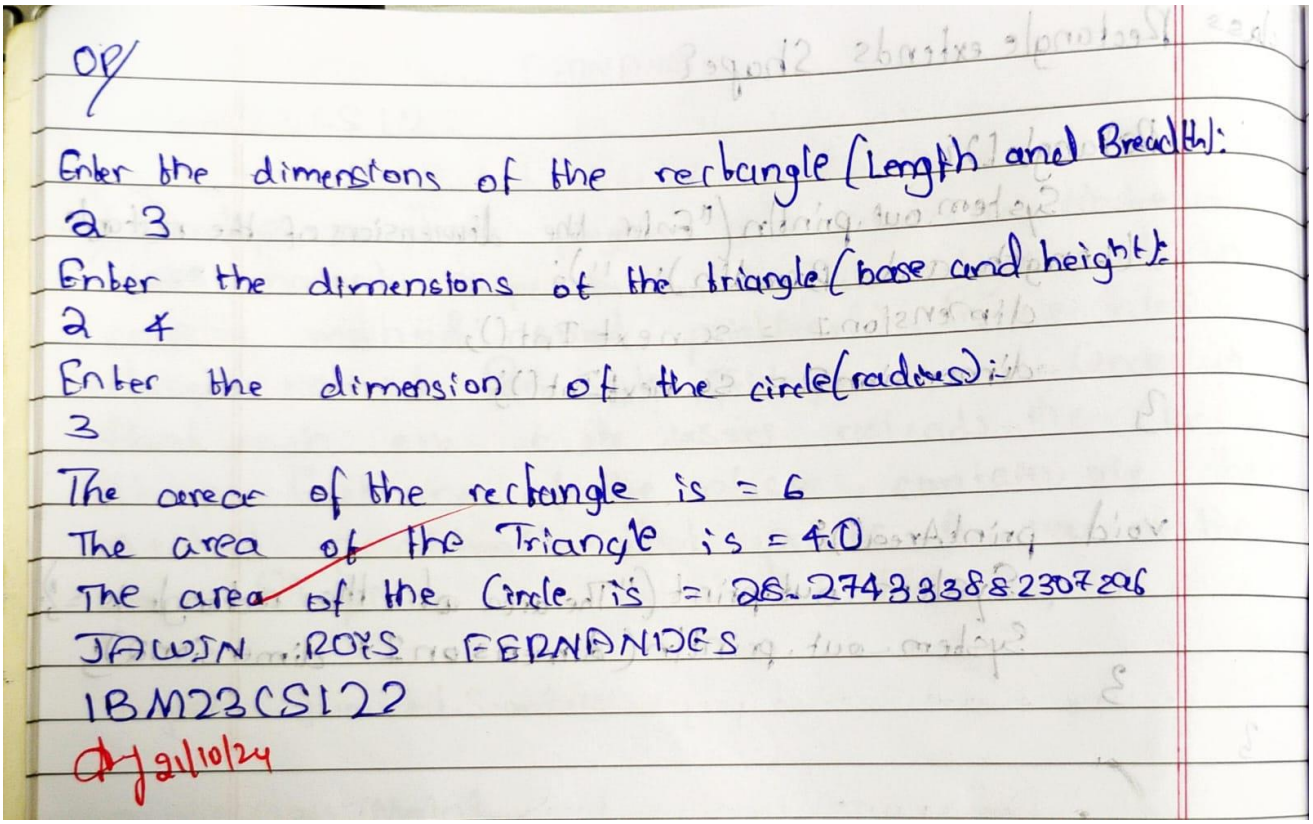
Algorithm:



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

```
import java.util.Scanner;

public class Main{
    public static void main(String[] args){
        Rectangle ob2 = new Rectangle();
        Triangle ob1 = new Triangle();
        Circle ob3 = new Circle();
        ob2.printArea();
        ob1.printArea();
        ob3.printArea();
        System.out.println("JAWIN ROYS FERNANDES");
        System.out.println("1BM23CS122");
    }
}

abstract class Shape{
    Scanner sc = new Scanner(System.in);
    int dimension1, dimension2;
    abstract void printArea();
}
```

```
class Rectangle extends Shape{

    Rectangle(){
        System.out.println("Enter the dimensions of the rectangle
        (Length and Breadth): ");
        dimension1 = sc.nextInt();
        dimension2 = sc.nextInt();
    }

    void printArea(){
        System.out.print("The area of the rectangle is =");
        System.out.println(dimension1 * dimension2);
    }
}

class Triangle extends Shape{
    Triangle(){
        System.out.println("Enter the dimensions of the
        triangle(base and height): ");
        dimension1 = sc.nextInt();
        dimension2 = sc.nextInt();
    }
    void printArea(){
        System.out.print("The are of the Triangle is = ");
        System.out.println(0.5* dimension1 * dimension2);
    }
}
class Circle extends Shape{
    Circle(){
        System.out.println("Enter the dimension of the circle(radius)");
        dimension1 = sc.nextInt();
    }
    void printArea(){
        System.out.print("The area of the Circle is = ");
        System.out.println(3.14159265358979*dimension1*dimension1);
    }
}
```



Code:

```
import java.util.Scanner ;

public class Main{
	public static void main(String[] args){
		Rectangle ob2  = new Rectangle() ;
		Triangle ob1 = new Triangle() ;
		Circle ob3 = new Circle() ;
		ob2.printArea() ;
		ob1.printArea() ;
		ob3.printArea() ;
		System.out.println("JAWIN ROYS FERNANDES" )  ;
		System.out.print("1BM23CS122") ;
	}
}

abstract class Shape{
	Scanner sc = new Scanner(System.in) ;
	int dimension1 , dimension2  ;
	abstract void printArea();
}


class Rectangle extends Shape{
```

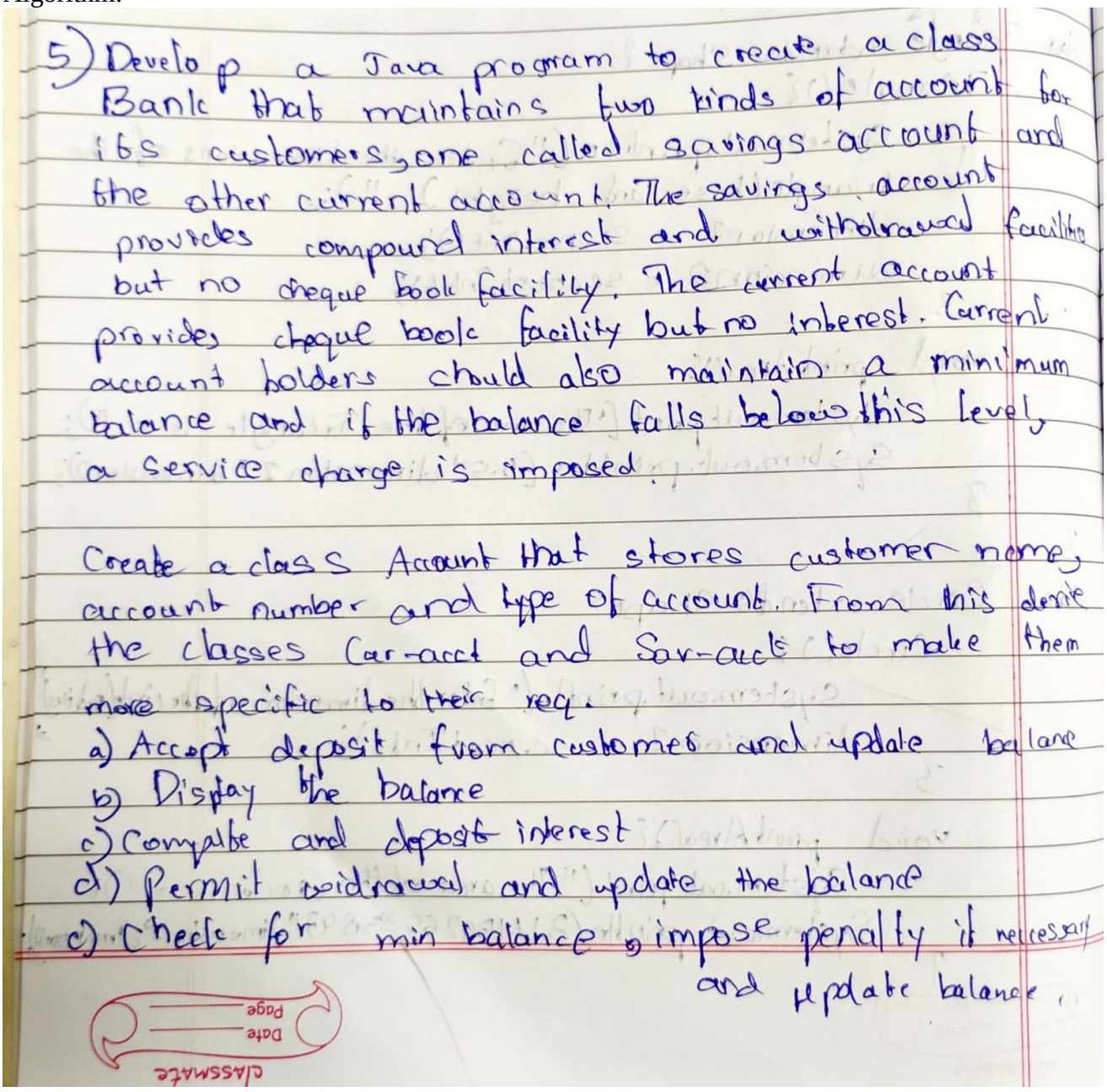
```
        Rectangle(){
                System.out.println("Enter the dimensions of the rectangle(Length and Breadth): " ) ;
                dimension1 = sc.nextInt() ;
                dimension2 = sc.nextInt() ;
        }

        void printArea(){
                System.out.print("The area of the rectangle is = ") ;
                System.out.println(dimension1*dimension2) ;
        }
}

class Triangle extends Shape{
        Triangle(){
                System.out.println("Enter the dimensions of the triangle(base and height): " ) ;
                dimension1 = sc.nextInt() ;
                dimension2 = sc.nextInt() ;
        }

        void printArea(){
                System.out.print("The area of the Triangle is = ") ;
                System.out.println(0.5*dimension1*dimension2) ;
        }
}
class Circle extends Shape{
        Circle(){
                System.out.println("Enter the dimension of the circle(radius): ")  ;
                dimension1 = sc.nextInt() ;

        }

        void printArea(){
                System.out.print("The area of the Circle is = ") ;
                System.out.println(3.1415926535897*dimension1*dimension1) ;
        }
}
```

```
Enter the dimensions of the rectangle(Length and Breadth):
2 3
Enter the dimensions of the triangle(base and height):
2 4
Enter the dimension of the circle(radius):
3
The area of the rectangle is = 6
The area of the Triangle is = 4.0
The area of the Circle is = 28.274333882307296
JAWIN ROYS FERNANDES
1BM23CS122
```

Program 5

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

Algorithm:



```
import java.util.Scanner;

public class Bank{
    static Scanner sc = new Scanner(System.in);
    Account obj;

    void createAccount(){
        String customer;
        int account;
        String type;
        int initBal;

        System.out.print("Enter the customer name: ");
        customer = sc.nextLine();
        System.out.print("Enter account Number: ");
        account = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter Account type (Savings or Current): ");
        type = sc.nextLine();
        System.out.print("Enter the initial Balance: ");
        initBal = sc.nextInt();

        if (type.equals("Savings")){
            obj = new Savings(customer, account, initBal);
        }
        else{
            obj = new Current(customer, account, initBal);
        }
    }

    public static void main(String[] args){
        Bank bank = new Bank();
        bank.createAccount();
```

```
while (true){
    System.out.println("- - - - - MENU - - - - -");
    System.out.println("1. Deposit  2. Withdraw");
    System.out.println("3. Compute interest");
    System.out.println("4. Display account details");
    System.out.println("5. exit");
    int choice = sc.nextInt();

    switch (choice){
        case 1:
            bank.obj.deposit();
            break;
        case 2:
            bank.obj.withdraw();
            break;
        case 3:
            if (bank.obj instanceof Savings){
                ((Savings)bank.obj).computeInterest();
            }
            else{
                System.out.println("Interest computation is
            only available for Savings accounts!");
            }
            break;
        case 4:
            bank.obj.display();
            break;
        case 5:
            break;
        default:
            System.out.println("Invalid choice. Please try again");
    }
    if(choice == 5) break;
}
}
```

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

Account(String customer, int accountNum, int bal){
    customerName = customer;
    accountNumber = accountNum;
    balance = bal;
}

void deposit(){
    System.out.print("Enter the amount to deposit:");
    int amt = Bank.sc.nextInt();
    balance += amt;
  System.out.println("Deposited:" + amt + ", NewBalance:" +
                     balance);
}

void withdraw(){
    System.out.print("Enter the amount to withdraw:");
    int amt = Bank.sc.nextInt();
    if (balance - amt < 0){
      System.out.println("Insufficient Balance to
          withdraw the given amount.");
    }else{
        balance -= amt;
      System.out.println("Amount of" + amt + "withdrawn
          successfully. Current Balance is" + balance);
    }
}

void display(){
    System.out.println("The Balance in the account is" +
        balance);
}
```

```
class Savings extends Account{
    double interestPercent;

    Savings(String customer, int accountNum, int bal){
        super(customer, accountNum, bal);
        System.out.print("Enter the interest percentage
        on the account: ");
        interestPercent = Bank.sc.nextDouble();
    }

    void computeInterest(){
        balance += balance*(interestPercent/100);
        System.out.println("Amount after applying
        interest is: " + balance);
    }
}

class Current extends Accounts{
    int minBalance = 1000;

    Current(String customer, int accountNum, int bal){
        super(customer, accountNum, bal);
    }

    void withdraw(){
        System.out.print("Enter the amount to withdraw");
        int amt = Bank.sc.nextInt();
        if(balance - amt < minBalance){
            System.out.println("Insufficient Balance to
            maintain the minimum required");
        }else{
            balance -= amt;
            System.out.println("Amount of " + amt +
            " withdrawn Successfully. Current Balance is " +
            balance);
        }
    }
```

Enter the customer name: Jawin
Enter account number : 122
Enter Account type (Savings or Current): Savings
Enter the initial Balance: 10000
Enter the interest percentage on the account: 6

---------- MENU ----------
1. Deposit 2. Withdraw
3. Compute Interest
4. Display account details.
5. exit
1
Enter the amount to deposit: 2000
Deposited: 2000, New Balance: 12000

---------- MENU ----------
1. Deposit 2. Withdraw
3. Compute interest
4. Display account details
5. exit
Enter the amount to withdraw 11900
Amount of 11900 withdrawn Successfully. Current Balance is 100

---------- MENU ----------
1. Deposit 2. Withdraw
3. Compute interest
4. Display account details
5. exit
4
The Balance in the account is 100

---------- MENU ----------
1. Deposit 2. Withdraw
3. Compute interest
4. Display account details
5. exit
5

Code:

```
import java.util.Scanner;

public class Bank {
   static Scanner sc = new Scanner(System.in);
   Account ob1;

   void createAccount() {
      String customer;
      int account;
      String type;
      int initBal;

      System.out.print("Enter the customer name: ");
      customer = sc.nextLine();
      System.out.print("Enter account Number: ");
      account = sc.nextInt();
      sc.nextLine();  // Consume the newline
      System.out.print("Enter Account type (Savings or Current): ");
      type = sc.nextLine();
      System.out.print("Enter the initial Balance: ");
      initBal = sc.nextInt();

      if (type.equals("Savings")) {
         ob1 = new Savings(customer, account, initBal);
      } else {
         ob1 = new Current(customer, account, initBal);
      }
   }

   public static void main(String[] args) {
      Bank bank = new Bank();
      bank.createAccount();

      while (true) {
         System.out.println("--------------------MENU-----------------");
         System.out.println("1. Deposit     2. Withdraw");
         System.out.println("3. Compute interest");
         System.out.println("4. Display account details");
         System.out.println("5. exit " ) ;
         int choice = sc.nextInt();

         switch (choice) {
            case 1:
               bank.ob1.deposit();
               break;
            case 2:
```

```
                bank.ob1.withdraw();
                break;
            case 3:
                if (bank.ob1 instanceof Savings) {
                    ((Savings) bank.ob1).computeInterest();
                } else {
                    System.out.println("Interest computation is only available for Savings accounts.");
                }
                break;
            case 4:
                bank.ob1.display();
                break;
                    case 5:
                            break ;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
                    if(choice == 5) break ;
    }
  }
}

class Account {
    String customerName;
    int accountNumber;
    int balance;

    Account(String customer, int accountNum, int bal) {
        customerName = customer;
        accountNumber = accountNum;
        balance = bal;
    }

    void deposit() {
        System.out.print("Enter the amount to deposit: ");
        int amt = Bank.sc.nextInt();
        balance += amt;
        System.out.println("Deposited: " + amt + ", New Balance: " + balance);
    }

    void withdraw() {
        System.out.print("Enter the amount to withdraw: ");
        int amt = Bank.sc.nextInt();
        if (balance - amt < 0) {
            System.out.println("Insufficient Balance to withdraw the given amount.");
        } else {
            balance -= amt;
```

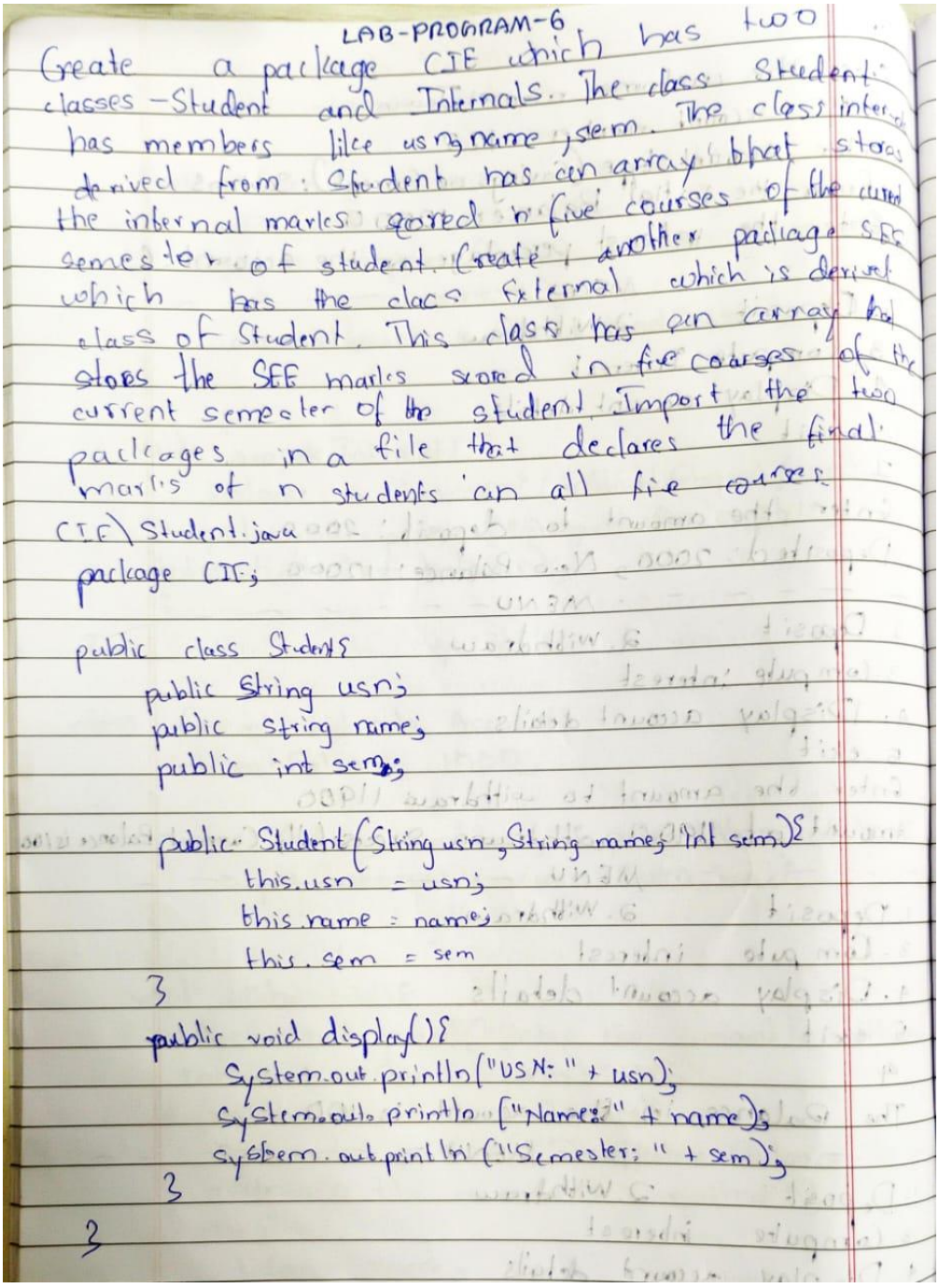
```
            System.out.println("Amount of " + amt + " withdrawn successfully. Current Balance is " +
balance);
        }
    }

    void display() {
        System.out.println("The Balance in the account is " + balance);
    }
}

class Savings extends Account {
    double interestPercent;

    Savings(String customer, int accountNum, int bal) {
        super(customer, accountNum, bal);
        System.out.print("Enter the interest percentage on the account: ");
        interestPercent = Bank.sc.nextDouble();
    }

    void computeInterest() {
        balance += balance * (interestPercent / 100);
        System.out.println("Amount after applying interest is: " + balance);
    }
}

class Current extends Account {
    int minBalance = 1000;

    Current(String customer, int accountNum, int bal) {
        super(customer, accountNum, bal);
    }

    void withdraw() {
        System.out.print("Enter the amount to withdraw: ");
        int amt = Bank.sc.nextInt();
        if (balance - amt < minBalance) {
            System.out.println("Insufficient Balance to maintain the minimum required.");
        } else {
            balance -= amt;
            System.out.println("Amount of " + amt + " withdrawn successfully. Current Balance is " +
balance);
        }
    }
}
```

```
PS D:\1BM23CS122> java Bank.java
Enter the customer name: Jawin
Enter account Number: 122
Enter Account type (Savings or Current): Savings
Enter the initial Balance: 10000
Enter the interest percentage on the account: 6
------------------MENU-----------------
1. Deposit     2. Withdraw
3. Compute interest
4. Display account details
5. exit
1
Enter the amount to deposit: 2000
Deposited: 2000, New Balance: 12000
------------------MENU-----------------
1. Deposit     2. Withdraw
3. Compute interest
4. Display account details
5. exit
2
Enter the amount to withdraw: 11900
Amount of 11900 withdrawn successfully. Current Balance is 100
------------------MENU-----------------
1. Deposit     2. Withdraw
3. Compute interest
4. Display account details
5. exit
4
The Balance in the account is 100
------------------MENU-----------------
1. Deposit     2. Withdraw
3. Compute interest
4. Display account details
5. exit
5
```

Program 6

Create a package CIE which has two classes - Personal 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 Personal. 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:



LAB-PROGRAM-6

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

CIE\Student.java

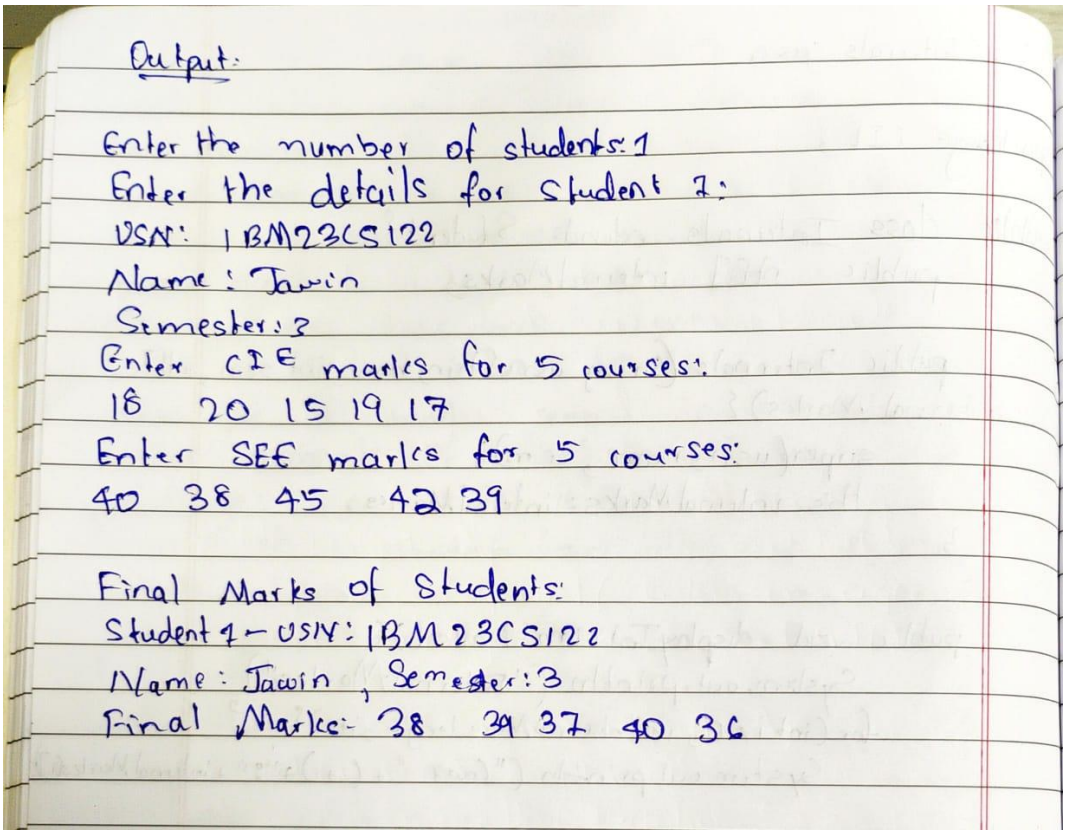
```
package CIE;

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

    public Student(String usn, String name, int sem){
        this.usn = usn;
        this.name = name;
        this.sem = sem
    }
    public void display(){
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
```

CIE/Internals.java

```
package CIE;

public class Internals extends Student{
    public int[] internalMarks;

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

    public void displayInternalMarks(){
        System.out.println("Internal Marks:");
        for(int i=0; i<internalMarks.length; i++){
            System.out.println("Course "+(i+1)+": "+internalMarks[i]);
```

SEE/External.java

```
package SEE;
import CIE.Student;

public class External extends Student{
    public int[] seeMarks;

    public External(String usn, String name, int sem, int[] seeMarks){
        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }
    public void displaySeeMarks(){
        System.out.println("SEE Marks:");
        for(int i=0; i<seeMarks.length; ++i){
            System.out.println("Course "+(i+1)+": "+seeMarks[i]);
        }
    }
}
```



Code:

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

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

        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 = scanner.next();
            System.out.print("Name: ");
            String name = scanner.next();
            System.out.print("Semester: ");
            int sem = scanner.nextInt();
```

```
            int[] internalMarks = new int[5];
            System.out.println("Enter internal marks for 5 courses:");
            for (int j = 0; j < 5; j++) {
                internalMarks[j] = scanner.nextInt();
            }
            internals[i] = new Internals(usn, name, sem, internalMarks);

            int[] seeMarks = new int[5];
            System.out.println("Enter SEE marks for 5 courses:");
            for (int j = 0; j < 5; j++) {
                seeMarks[j] = scanner.nextInt();
            }
            externals[i] = new External(usn, name, sem, seeMarks);
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            internals[i].display();
            internals[i].displayInternalMarks();
            externals[i].displaySeeMarks();

            System.out.println("Final Marks: ");
            for (int j = 0; j < 5; j++) {
                int finalMarks = internals[i].internalMarks[j] + externals[i].seeMarks[j];
                System.out.println("Course " + (j + 1) + ": " + finalMarks);
            }
            System.out.println();
        }

        scanner.close();
    }
}
package CIE;

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

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

    public void display() {
        System.out.println("USN: " + usn);
```

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

```
package CIE;

public class Internals extends Student {
    public int[] internalMarks;

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

    public void displayInternalMarks() {
        System.out.println("Internal Marks: ");
        for (int i = 0; i < internalMarks.length; i++) {
            System.out.println("Course " + (i + 1) + ": " + internalMarks[i]);
        }
    }
}
```

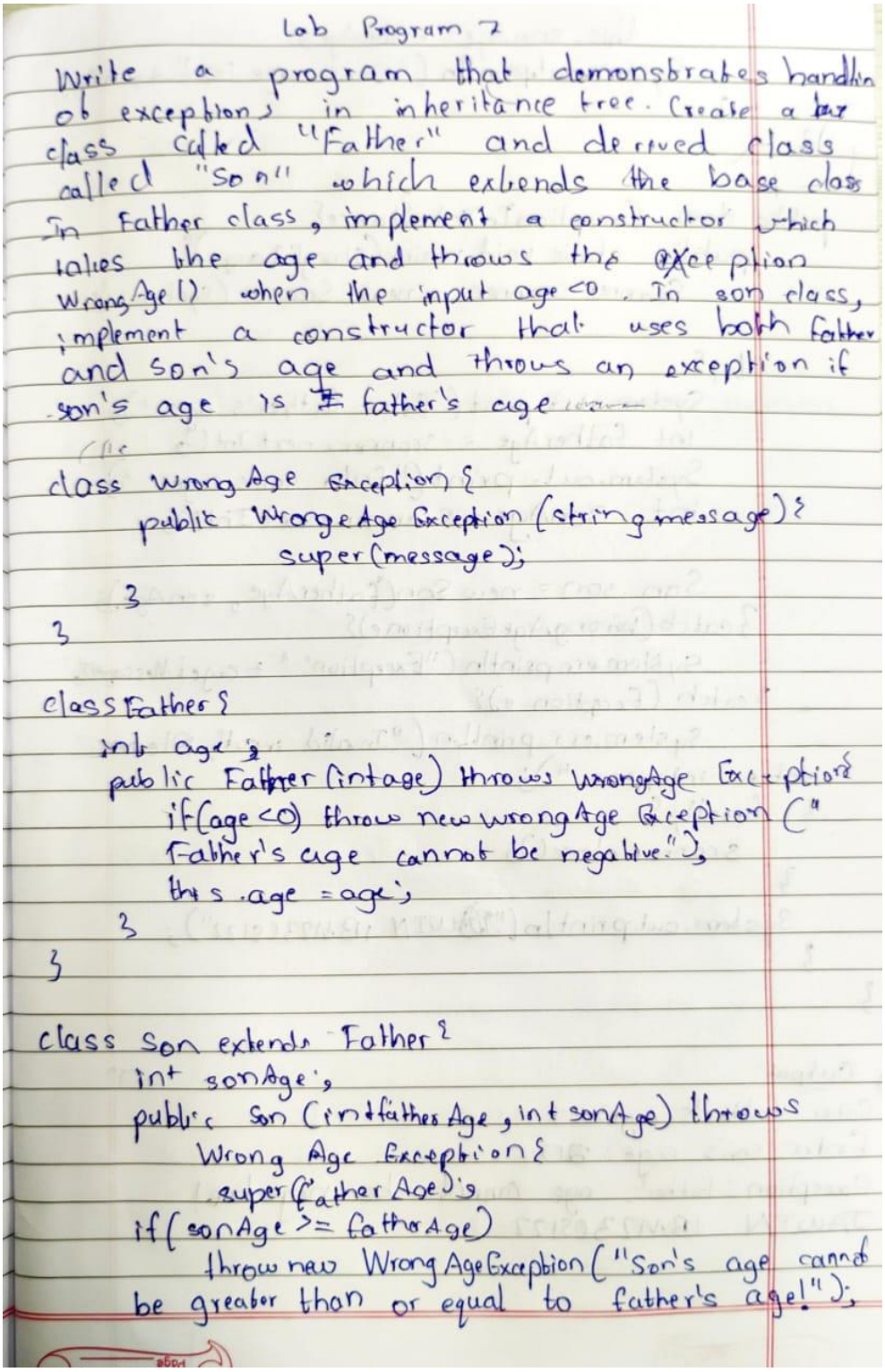
```
Final Marks of Students:
USN: 122
Name: Jawin
Semester: 5
Internal Marks:
Course 1: 29
Course 2: 30
Course 3: 36
Course 4: 37
Course 5: 35
SEE Marks:
Course 1: 45
Course 2: 46
Course 3: 44
Course 4: 49
Course 5: 50
Final Marks:
Course 1: 74
Course 2: 76
Course 3: 80
Course 4: 86
Course 5: 85

USN: 119
Name: Irfan
Semester: 5
Internal Marks:
Course 1: 35
Course 2: 37
Course 3: 36
Course 4: 30
Course 5: 29
SEE Marks:
Course 1: 50
Course 2: 49
Course 3: 44
Course 4: 46
Course 5: 45
Final Marks:
Course 1: 85
Course 2: 86
Course 3: 80
Course 4: 76
Course 5: 74
```

```
PS D:\1BM23CS122\package> java Main.java
Enter the number of students: 2
Enter details for student 1:
USN: 122
Name: Jawin
Semester: 5
Enter internal marks for 5 courses:
29
30
36
37
35
Enter SEE marks for 5 courses:
45
46
44
49
50
Enter details for student 2:
USN: 119
Name: Irfan
Semester: 5
Enter internal marks for 5 courses:
35
37
36
30
29
Enter SEE marks for 5 courses:
50
49
44
46
45
```

Program 7

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=father’s age.

Algorithm:



Lab Program 7

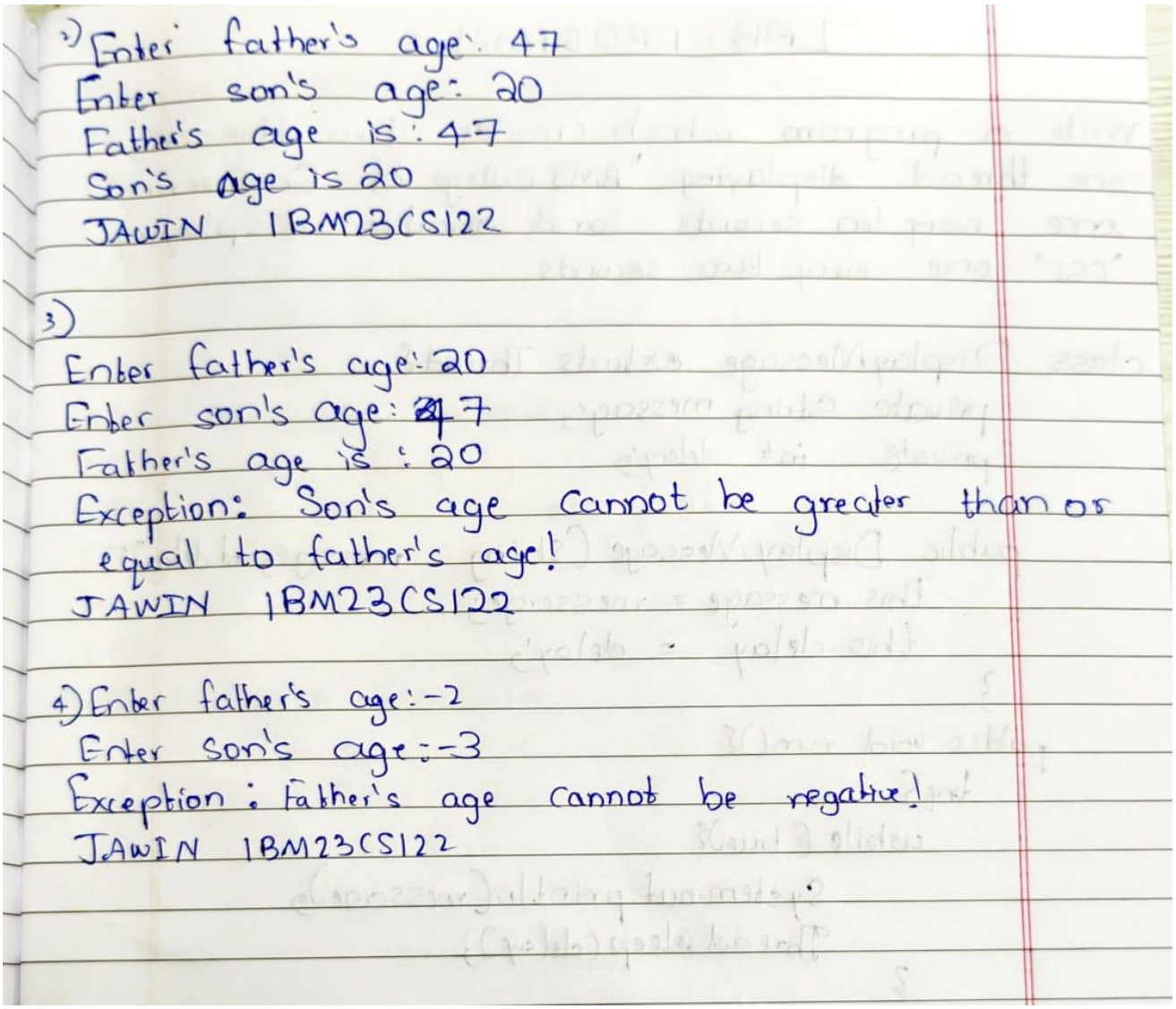
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age<0. In son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

```
class WrongAgeException {
    public WrongAgeException(String message) {
        super(message);
    }
}

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

class Son extends Father {
    int sonAge;
    public Son(int fatherAge, int sonAge) throws
        WrongAgeException {
        super(fatherAge);
    if(sonAge >= fatherAge)
        throw new WrongAgeException("Son's age cannot
    be greater than or equal to father's age!");
```

```
        this.sonAge = sonAge;
        System.out.println("Son's age is: " + sonAge);
    }
}

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

        try{
            System.out.print("Enter father's age: ");
            int fatherAge = scanner.nextInt();
            System.out.print("Enter son's age: ");
            int sonAge = scanner.nextInt();

            Son son = new Son(fatherAge, sonAge);
        }catch(WrongAgeException e){
            System.err.println("Exception: " + e.getMessage());
        }catch (Exception e){
            System.err.println("Invalid input! Please
enter integers.");
        } finally{
            scanner.close();
        }
        System.out.println("JAWIN IBM23CS122");
    }
}
```

Output

```
Enter father's age: -1
Enter son's age: 20
Exception: Father's age cannot be negative!
JAWIN IBM23CS122
```



Code:

```
import java.util.Scanner;

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

class Father {
    protected int fatherAge;

    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Father's age cannot be negative!");
```

```
        }
        this.fatherAge = age;
        System.out.println("Father's age is: " + fatherAge);
    }
}

class Son extends Father {
    private int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be negative!");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to father's age!");
        }
        this.sonAge = sonAge;
        System.out.println("Son's age is: " + sonAge);
    }
}

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

        try {
            System.out.print("Enter father's age: ");
            int fatherAge = scanner.nextInt();
            System.out.print("Enter son's age: ");
            int sonAge = scanner.nextInt();

            Son son = new Son(fatherAge, sonAge);
        } catch (WrongAgeException e) {
            System.err.println("Exception: " + e.getMessage());
        } catch (Exception e) {
            System.err.println("Invalid input! Please enter integers.");
        } finally {
            scanner.close();
        }
        System.out.println("JAWIN 1BM23CS122") ;
```

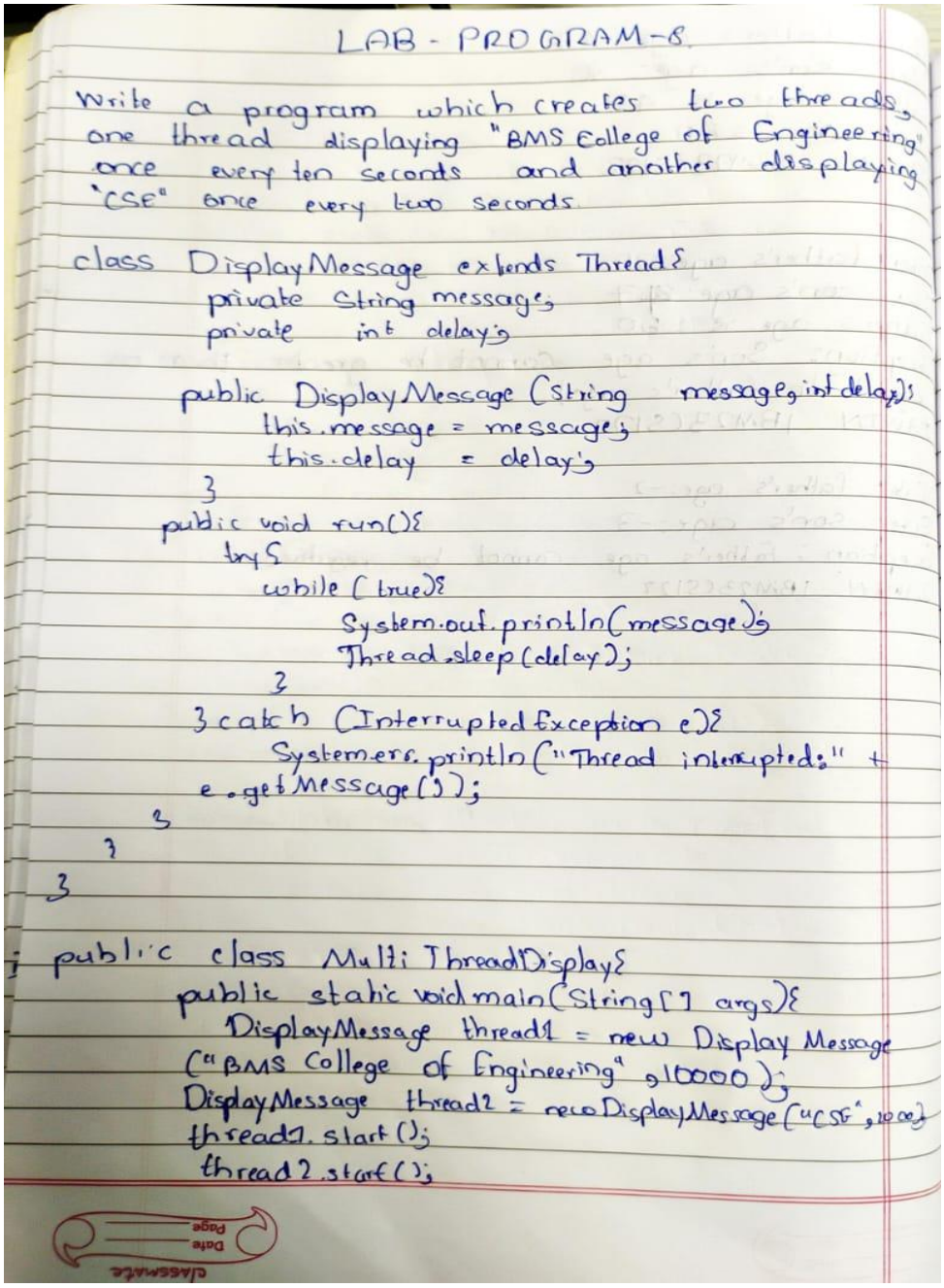
```
    }
}
```

```
PS C:\javaCSE\program7> javac .\ExceptionInInheritance.java
PS C:\javaCSE\program7> java .\ExceptionInInheritance.java
Enter father's age: -1
Enter son's age: 20
Exception: Father's age cannot be negative!
JAWIN 1BM23CS122
PS C:\javaCSE\program7> 47
47
PS C:\javaCSE\program7> java .\ExceptionInInheritance.java
Enter father's age: 47
Enter son's age: 20
Father's age is: 47
Son's age is: 20
JAWIN 1BM23CS122
PS C:\javaCSE\program7> java .\ExceptionInInheritance.java
Enter father's age: 20
Enter son's age: 47
Father's age is: 20
Exception: Son's age cannot be greater than or equal to father's age!
JAWIN 1BM23CS122
PS C:\javaCSE\program7> java .\ExceptionInInheritance.java
Enter father's age: -2
Enter son's age: -3
Exception: Father's age cannot be negative!
JAWIN 1BM23CS122
PS C:\javaCSE\program7>
```

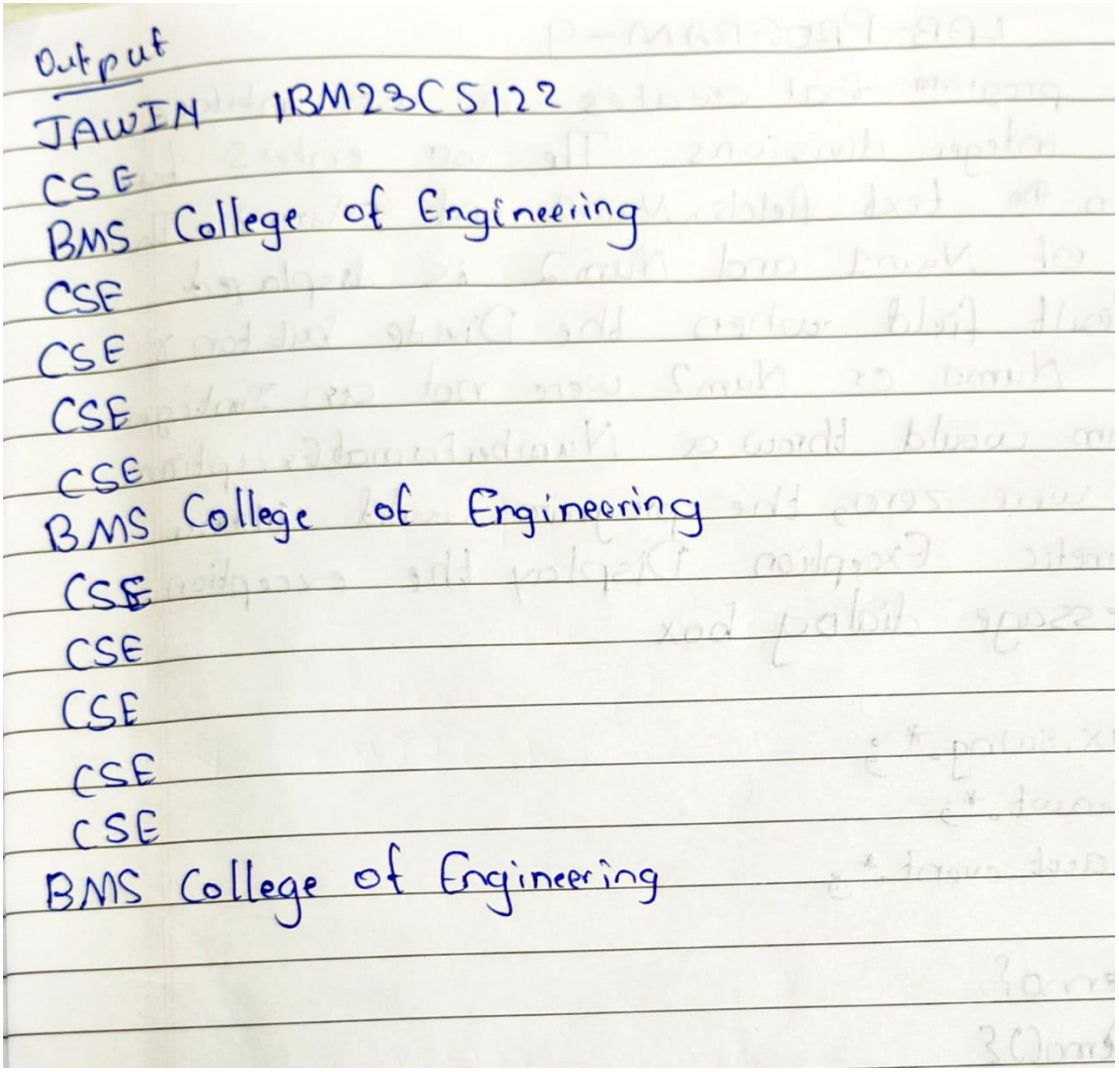
Program 8

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

Algorithm:



```
LAB - PROGRAM-8.

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

class DisplayMessage extends Thread{
        private String message;
        private int delay;

        public DisplayMessage (String message, int delay){
            this.message = message;
            this.delay = delay;
        }
      public void run(){
          try{
              while (true){
                  System.out.println(message);
                  Thread.sleep(delay);
              }
          }catch (InterruptedException e){
              System.err.println("Thread interrupted:" +
          e.getMessage());
      }
  }
}

public class MultiThreadDisplay{
      public static void main(String[] args){
        DisplayMessage thread1 = new DisplayMessage
        ("BMS College of Engineering", 10000);
        DisplayMessage thread2 = new DisplayMessage("CSE", 2000);
        thread1.start();
        thread2.start();
```



Code:

```
class DisplayMessage extends Thread {
    private String message;
    private int delay;

    public DisplayMessage(String message, int delay) {
        this.message = message;
        this.delay = delay;
    }

    public void run() {
        try {
            while (true) {
                System.out.println(message);
                Thread.sleep(delay);
```

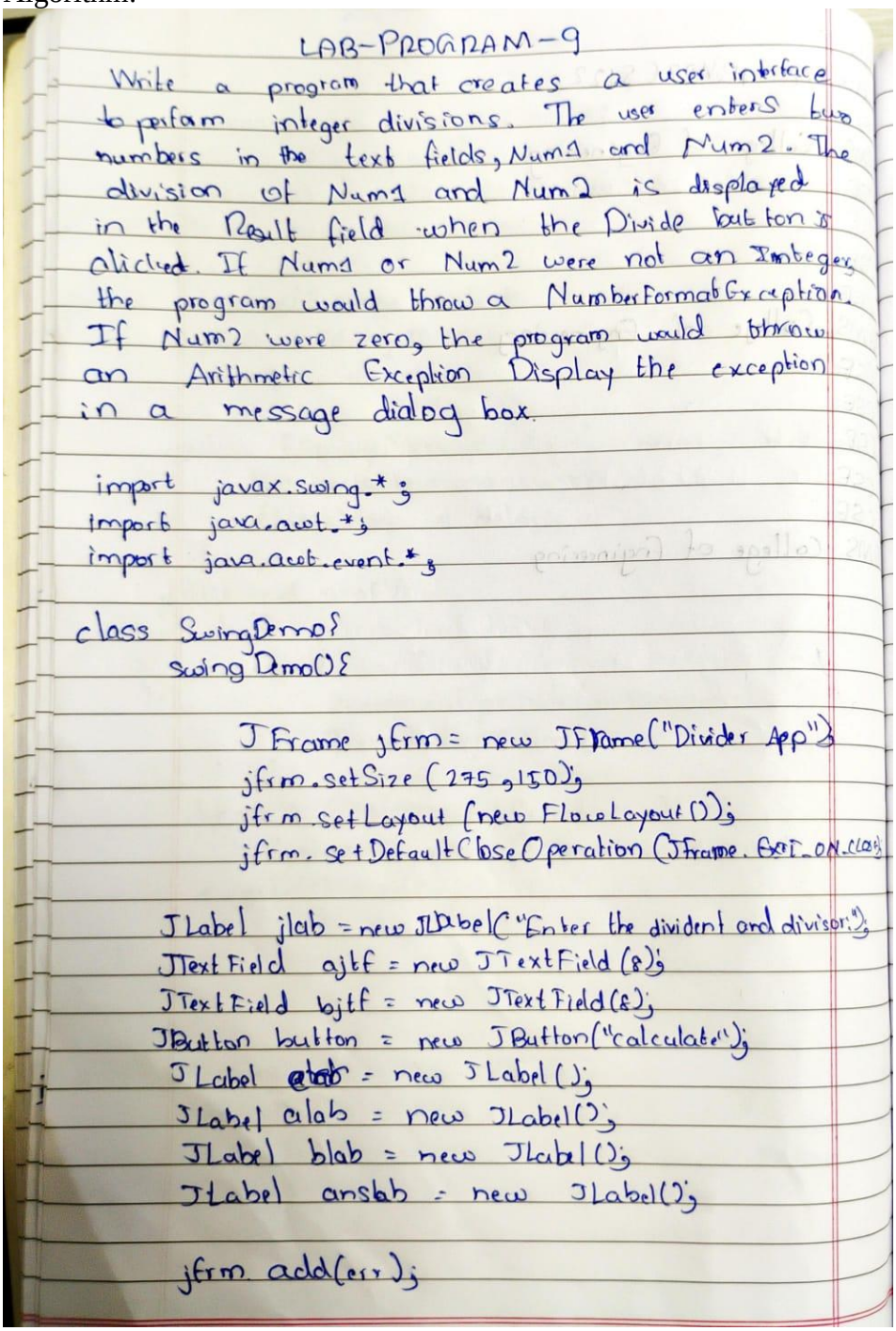
```
        }
      } catch (InterruptedException e) {
        System.err.println("Thread interrupted: " + e.getMessage());
      }
    }
}

public class MultiThreadDisplay {
    public static void main(String[] args) {
      DisplayMessage thread1 = new DisplayMessage("BMS College of Engineering", 10000);
      DisplayMessage thread2 = new DisplayMessage("CSE", 2000);
      thread1.start();
      thread2.start();
      System.out.println("JAWIN 1BM23CS122") ;
    }
}
```

```
PS C:\javaCSE\program8> javac .\MultiThreadDisplay.java
PS C:\javaCSE\program8> java .\MultiThreadDisplay.java
JAWIN 1BM23CS122
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
PS C:\javaCSE\program8>
```

Program 9

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

Algorithm:

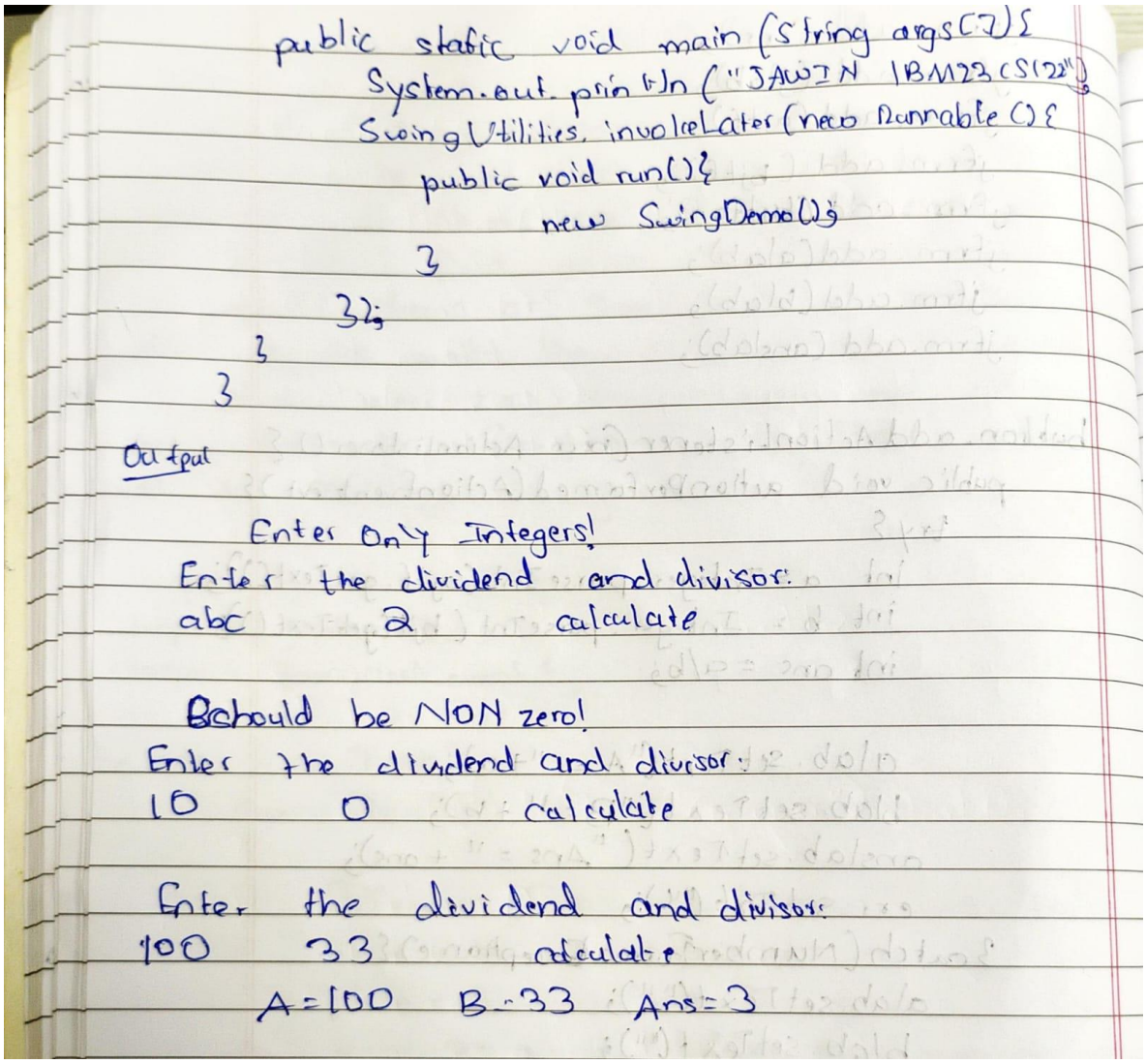


LAB-PROGRAM-9

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

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

class SwingDemo{
    SwingDemo(){

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

    JLabel jlab = new JLabel("Enter the divident and divisor:");
    JTextField ajtf = new JTextField(8);
    JTextField bjtf = new JTextField(8);
    JButton button = new JButton("calculate");
    JLabel elab = new JLabel();
    JLabel alab = new JLabel();
    JLabel blab = new JLabel();
    JLabel anslab = new JLabel();

    jfrm.add(err);
```

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

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

            alab.setText("A = "+a);
            blab.setText("B = "+b);
            anslab.setText("Ans = "+ans);
            err.setText("");
        } catch(NumberFormatException e) {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter only Integers!");
        } catch(ArithmeticException e) {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("B should be NON zero!");
        }
    }
});
jfrm.setVisible(true);
}
```



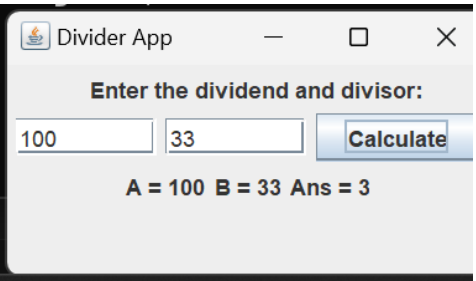
Code:

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

class SwingDemo {
   SwingDemo() {

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


      JLabel jlab = new JLabel("Enter the dividend and divisor:");
      JTextField ajtf = new JTextField(8);
```

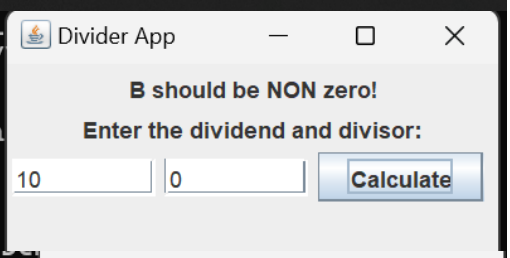
```
    JTextField bjtf = new JTextField(8);
    JButton button = new JButton("Calculate");
    JLabel err = new JLabel();
    JLabel alab = new JLabel();
    JLabel blab = new JLabel();
    JLabel anslab = new JLabel();


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


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

                alab.setText("A = " + a);
                blab.setText("B = " + b);
                anslab.setText("Ans = " + ans);
                err.setText("");
            } catch (NumberFormatException e) {
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText("Enter Only Integers!");
            } catch (ArithmeticException e) {
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText("B should be NON zero!");
            }
        }
    });


    jfrm.setVisible(true);
}
```

```
    public static void main(String args[]) {
        System.out.println("JAWIN 1BM23CS122") ;
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new SwingDemo();
            }
        });
    }
}
```

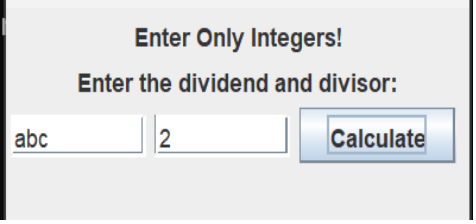
```
PS C:\javaCSE\program9> java .\SwingDemo.java
PS C:\javaCSE\program9> javac .\SwingDemo.java
PS C:\javaCSE\program9> java .\SwingDemo.java
JAWIN 1BM23CS122
```




```
Error: Could not find or load main class .\Swi
Caused by: java.lang.ClassNotFoundException: /
PS C:\javaCSE\program9> java .\SwingDemo.java
PS C:\javaCSE\program9> javac .\SwingDemo.java
PS C:\javaCSE\program9> java .\SwingDemo.java
JAWIN 1BM23CS122
```




```
Caused by: java.lang.ClassNotFoundException: /\Swi
PS C:\javaCSE\program9> java .\SwingDemo.java
PS C:\javaCSE\program9> javac .\SwingDemo.java
PS C:\javaCSE\program9> java .\SwingDemo.java
JAWIN 1BM23CS122
```

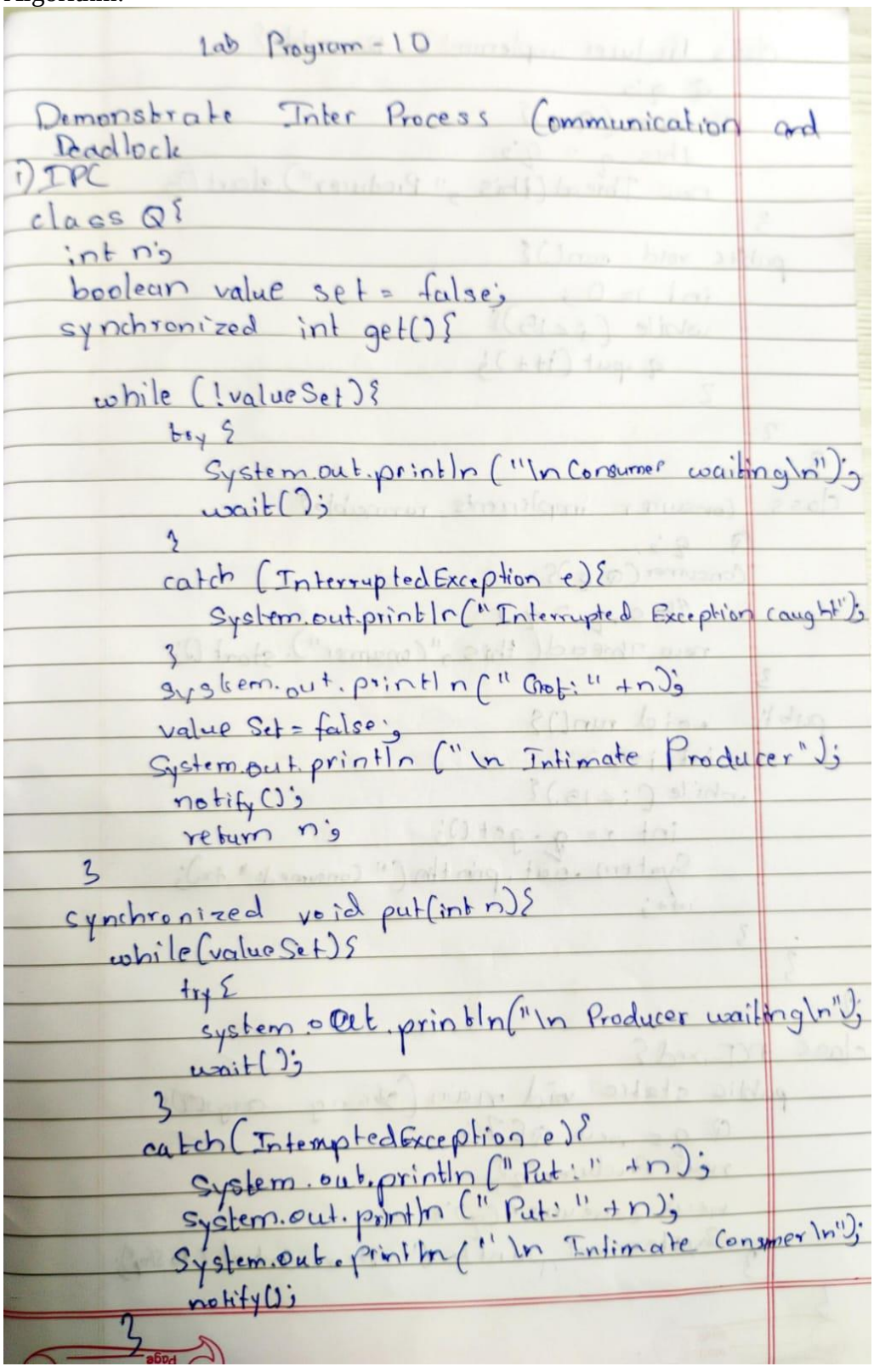


Program 10

Demonstrate Inter process Communication and deadlock

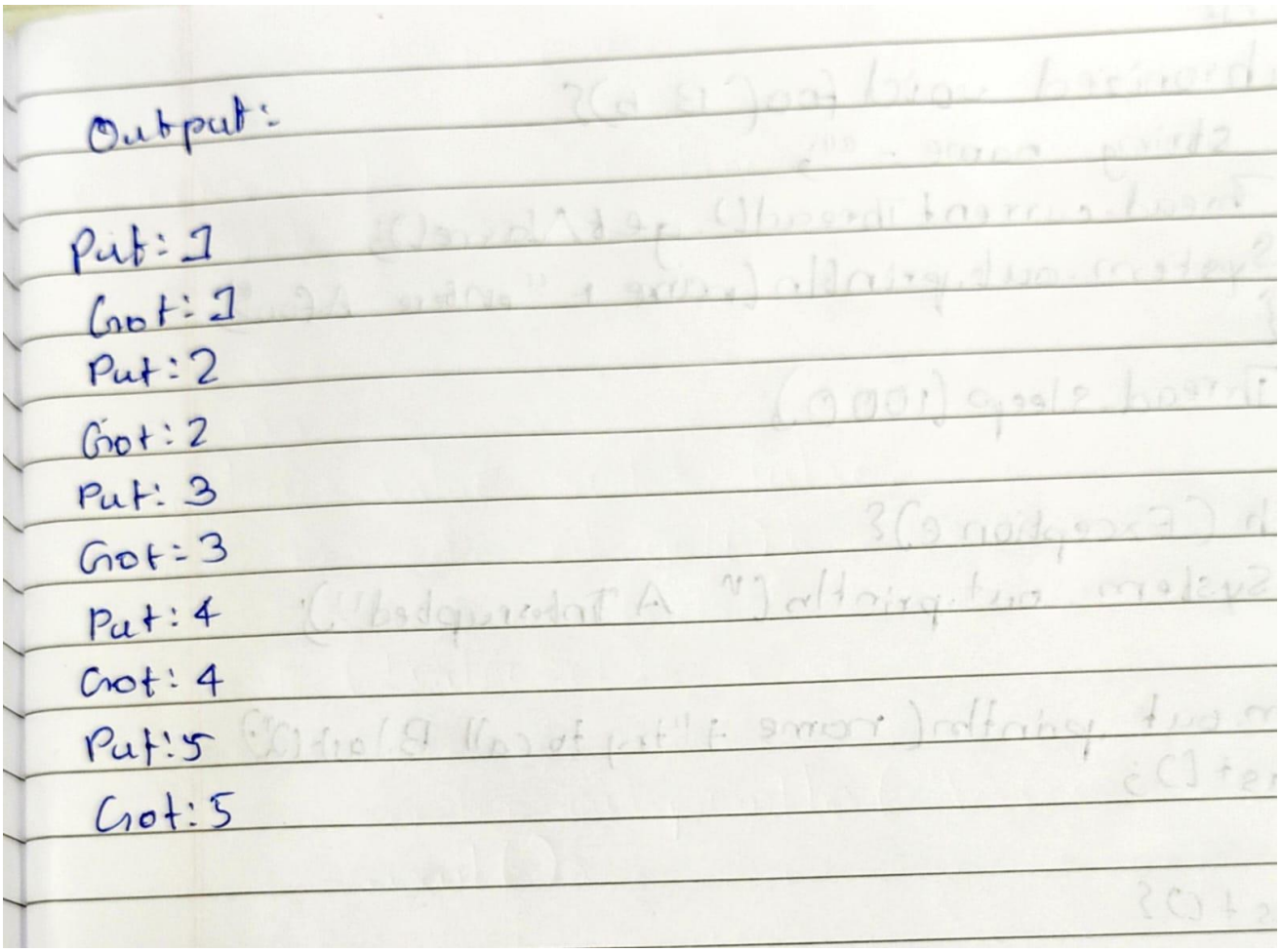
IPC

Algorithm:



```
Lab Program - 10

Demonstrate Inter Process Communication and
Deadlock
i) IPC
class Q{
  int n;
  boolean valueSet = false;
  synchronized int get(){

    while (!valueSet){
        try {
          System.out.println("\n Consumer waiting\n");
          wait();
        }
        catch (InterruptedException e){
          System.out.println("Interrupted Exception caught");
        }
        System.out.println(" Got: " +n);
        valueSet = false;
        System.out.println("\n Intimate Producer");
         notify();
         return n;
  }
synchronized void put(int n){
   while(valueSet){
       try{
         system.out.println("\n Producer waiting\n");
         wait();
      }
      catch(InterruptedException e){
         System.out.println(" Put: " +n);
         System.out.println(" Put: " +n);
         System.out.println("\n Intimate Consumer\n");
         notify();
      }
```

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



Code:

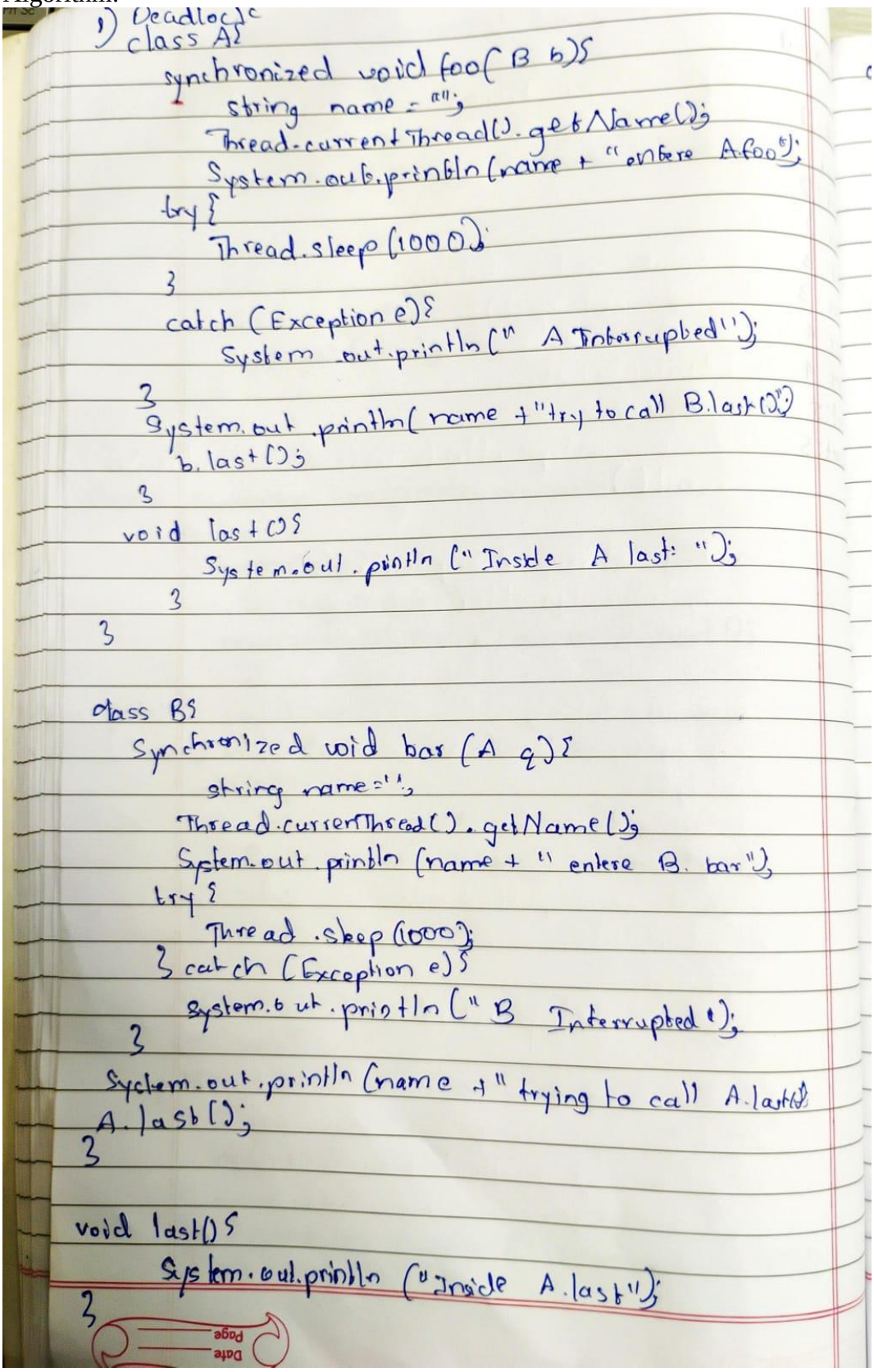
```
class Q {
    int n;
    boolean valueSet = false;

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

```
        return n;
    }

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

class Producer implements Runnable {
    Q q;

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

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

class Consumer implements Runnable {
    Q q;

    Consumer(Q q) {
        this.q = q;
```

```
        new Thread(this, "Consumer").start();
    }

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

public class PCFixed {
    public static void main(String args[]) {
        System.out.println("JAWIN 1BM23CS122") ;
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
            System.out.println("JAWIN 1BM23CS122");
    }
}
```

```
PS C:\javaCSE\program10\IPC> java .\PCFixed.java
Press Control-C to stop.
Put: 0
Notifying Consumer...
Producer waiting...
Got: 0
Notifying Producer...
Put: 1
Notifying Consumer...
Producer waiting...
Consumed: 0
Got: 1
Notifying Producer...
Consumed: 1
Put: 2
Notifying Consumer...
Producer waiting...
Got: 2
Notifying Producer...
Consumed: 2
Put: 3
Notifying Consumer...
Producer waiting...
Got: 3
Notifying Producer...
Consumed: 3
Put: 4
Notifying Consumer...
Producer waiting...
Got: 4
Notifying Producer...
Consumed: 4
Put: 5
Notifying Consumer...
Producer waiting...
Got: 5
Notifying Producer...
Consumed: 5
Put: 6
Notifying Consumer...
Producer waiting...
Got: 6
Notifying Producer...
Consumed: 6
Put: 7
Notifying Consumer...
Producer waiting...
Got: 7
Notifying Producer...
Consumed: 7
Put: 8
Notifying Consumer...
Producer waiting...
Got: 8
Notifying Producer...
Consumed: 8
Put: 9
Notifying Consumer...
Producer waiting...
Got: 9
Notifying Producer...
Consumed: 9
Put: 10
Notifying Consumer...
Producer waiting...
Got: 10
Notifying Producer...
Consumed: 10
```

DeadLock
Algorithm:



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

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

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

```
class Deadlock implements Runnable{
    A a = new A();
    B b = new B();
    Deadlock(){
        Thread.currentThread().setName("Main Thread");
        Thread t = new Thread(this, "Racing thread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }

    public void run(){
        b.bar(a);
        System.out.println("Back in other thread");
    }
    public static void main(String args[]){
        new Deadlock();
    }
}
```

Output:

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

Code:

```
class A {

   synchronized void foo(B b) {
      String name = Thread.currentThread().getName();

      System.out.println(name + " entered A.foo");

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

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

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

class B {

   synchronized void bar(A a) {
      String name = Thread.currentThread().getName();

      System.out.println(name + " entered B.bar");

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

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

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

class Deadlock implements Runnable {

    A a = new A();
    B b = new B();

    Deadlock() {
        Thread.currentThread().setName("MainThread");

        Thread t = new Thread(this, "RacingThread");
        t.start();

        a.foo(b); // get lock on a in this thread.

        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a); // get lock on b in other thread.

        System.out.println("Back in other thread");
    }

    public static void main(String args[]) {
        new Deadlock();
            System.out.println("JAWIN 1BM23CS122");
    }
}
```

```
PS C:\javaCSE\program10\Deadlock> javac .\Deadlock.java
PS C:\javaCSE\program10\Deadlock> java .\Deadlock.java
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
MainThread trying to call B.last()
Inside A.last
Inside B.last
Back in main thread
JAWIN 1BM23CS122
Back in other thread
```