VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Meghana U (24BECS401)

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 **Meghana U (24BECS401),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

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

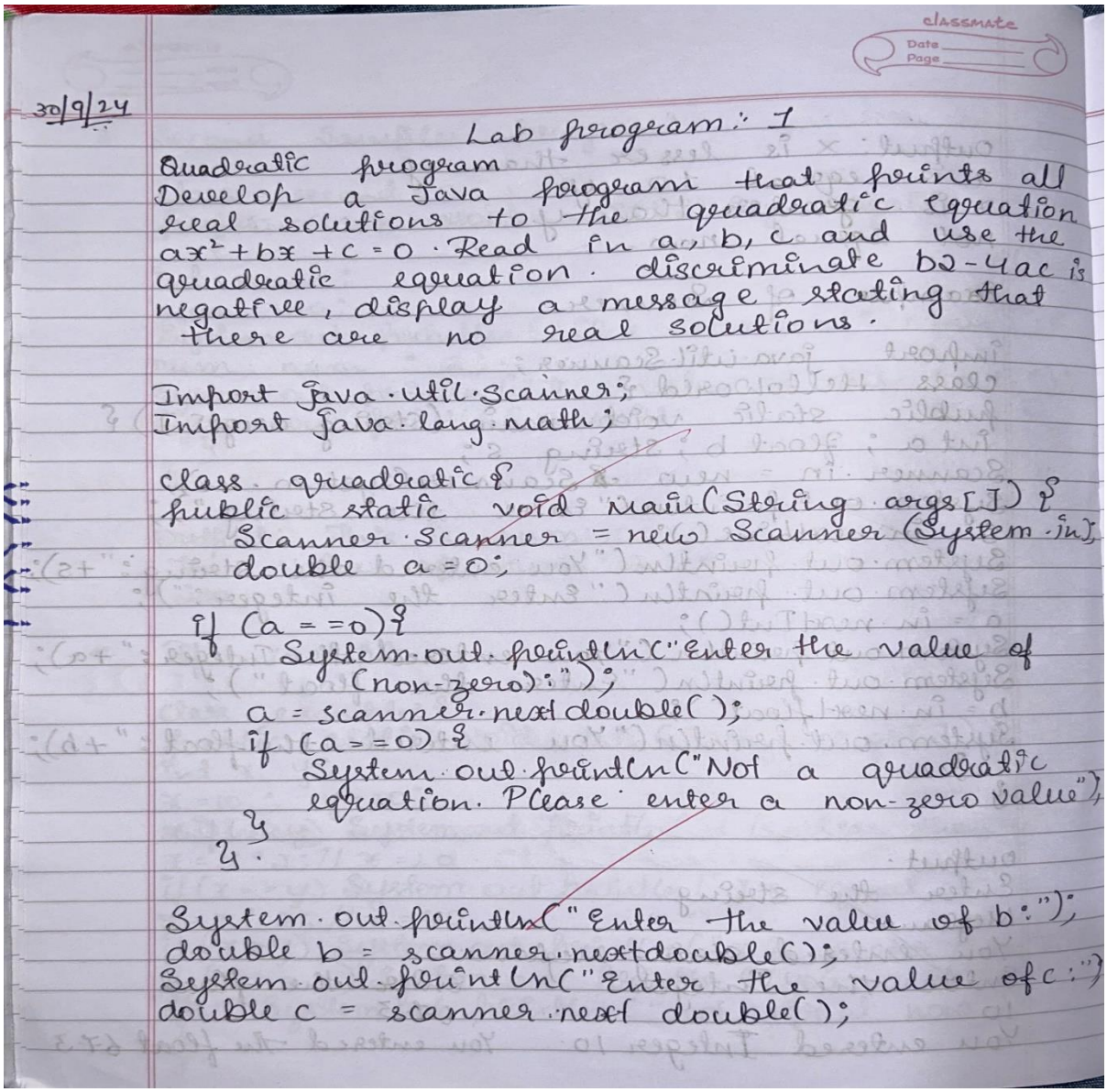
Index

Github Link:
https://github.com/MeghanaaU/java_actualprograms/tree/main

Program 1

Implement Quadratic Equation

Algorithm:



30/9/24

Lab program: 1

Quadratic program
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 equation. discriminate b^2-4ac is negative, display a message stating that there are no real solutions.

```
Import java.util.Scanner;
Import java.lang.math;

class quadratic{
public static void main(String args[]){
    Scanner scanner = new Scanner(System.in);
    double a=0;

if (a==0){
    System.out.println("Enter the value of (non-zero):");
    a = scanner.nextdouble();
    if (a==0){
        System.out.println("Not a quadratic equation. Please enter a non-zero value");
}
}.

System.out.println("Enter the value of b:");
double b = scanner.nextdouble();
System.out.println("Enter the value of c:")
double c = scanner.next double();
```

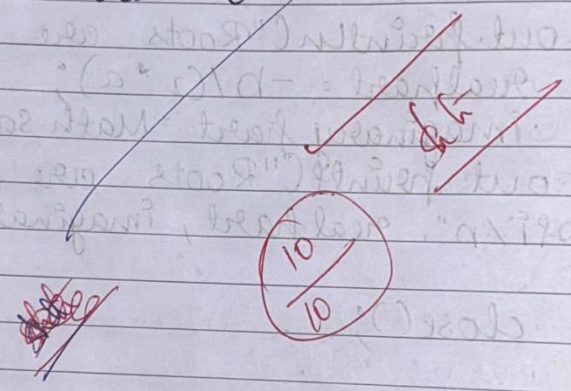
```
double d = (b * b - 4 * a * c);
if (d == 0) {
   double r1 = (-b) / (2 * a);
   System.out.printf("Roots are real & equal
   : %.2f & %.2f\n", r1, r1);
   }
else if (d > 0) {
   double r1 = (-b + Math.sqrt(d)) / (2 * a);
   double r2 = (-b - Math.sqrt(d)) / (2 * a);
   System.out.printf("The roots are:
      %.2f & %.2f%n", r1, r2);
else {
   System.out.println("Roots are imaginary");
   double realpart = -b / (2 * a);
   double imaginarypart = Math.sqrt(-d) / 2 * a;
   System.out.printf("Roots are: %.2f ±
      %.2fi%n", realpart, imaginarypart);
   }
   scanner.close();
   }
}
```

Output:

```
Enter the value of non zero.
10
Enter the value of b:
-13
Enter the value of c
-19
The roots are 2.17 & -0.87.
```

3)

Enter the value of a non-zero.
0
Not a quadratic equation. Please enter
a non-zero value.
Enter the value of a
1
Enter the value of b
1
Enter the value of c
2
Roots are imaginary
Roots are: -0.50 ± 1.32i.



Source Code:

```
import java.util.Scanner;
import java.lang.Math;

class quadratic {
    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);
        double a = 0;


        if (a == 0) {
            System.out.println("Enter the value of a (non-zero):");
            a = scanner.nextDouble();
            if (a == 0) {
                System.out.println("Not a quadratic equation. Please enter a non-zero value.");
            }
        }

        System.out.println("Enter the value of b:");
        double b = scanner.nextDouble();
        System.out.println("Enter the value of c:");
        double c = scanner.nextDouble();

        double d = (b * b - 4 * a * c);
        if (d == 0) {
            double r1 = (-b) / (2 * a);
            System.out.printf("Roots are real and equal: %.2f and %.2f%n", r1, r1);
        }
         else if (d > 0) {
            double r1 = (-b + Math.sqrt(d)) / (2 * a);
            double r2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.printf("The roots are: %.2f and %.2f%n", r1, r2);
        }
         else {
            System.out.println("Roots are imaginary.");
            double realPart = -b / (2 * a);
            double imaginaryPart = Math.sqrt(-d) / (2 * a);
            System.out.printf("Roots are: %.2f ± %.2fi%n", realPart, imaginaryPart);
        }

        scanner.close();


    }
}
```

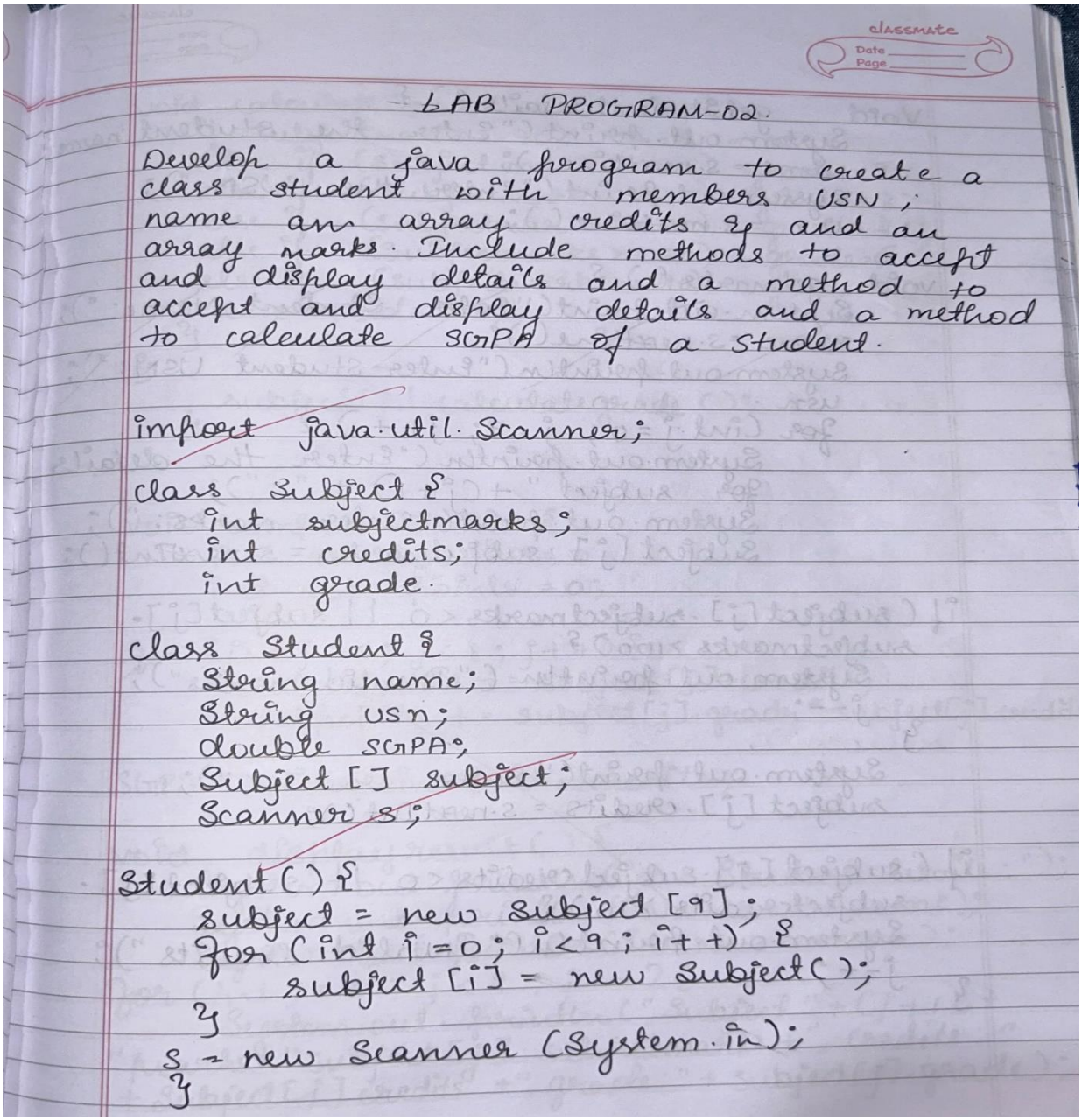
Output:

```
D:\24BECS401>java quadratic
Enter the value of a (non-zero):
-23
Enter the value of b:
-24
Enter the value of c:
-90
Roots are imaginary.
Roots are: -0.52 ± -1.91i

D:\24BECS401>java quadratic
Enter the value of a (non-zero):
1
Enter the value of b:
1
Enter the value of c:
2
Roots are imaginary.
Roots are: -0.50 ± 1.32i

D:\24BECS401>java quadratic
Enter the value of a (non-zero):
10
Enter the value of b:
-13
Enter the value of c:
-19
The roots are: 2.17 and -0.87

D:\24BECS401>java quadratic
Enter the value of a (non-zero):
0
Not a quadratic equation. Please enter a non-zero value.
Enter the value of a (non-zero):
1
Enter the value of b:
1
Enter the value of c:
2
Roots are imaginary.
Roots are: -0.50 ± 1.32i
```

Program 2

SGPA Calculation

Algorithm:



LAB PROGRAM-02.

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 accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;

class Subject {
    int subjectmarks;
    int credits;
    int grade.

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

Student() {
    subject = new Subject [9];
    for (int i=0; i<9; i++) {
        subject [i] = new Subject();
    }
  s = new Scanner (System.in);
  }
```

```
Void getStudentdetails () {
    System.out.print("Enter the student name:");
    name = s.nextline();
    System.out.print("Enter the USN:");
    usn = s.nextline();

void getmarks () {
    System.out.print("Enter Student Name:");
    name = s.nextline();
    System.out.println("Enter Student USN:");
    usn.
    for (int j = 0; j = 9; j++) {
        System.out.println("Enter the details
        for subject " + (j+1) + ":");
        System.out.print("Enter marks:");
        Subject[j].subjectmarks = s.nextInt();

if (subject[j].subjectmarks < 0 || subject[j].
  subjectmarks > 100) {
    System.out.println("Invalid marks");
    j--;
}
    System.out.print("Enter credits:");
    subject[j].credits = s.nextInt();

if (subject[j].subjectcredits < 0 || subject[j].
  subjectcredits > 10) {
    System.out.println("Invalid credits");
    j--;
}
```

```
void calculateGrade() {
    if subjectmarks >=90) grade = 10.
    else if (subject marks >=80) grade = 9;
    else if (subjectmarks >=70) grade = 8;
    else if (subjectmarks >= 60) grade = 7;
    else if (subjectmarks >= 50) grade = 6;
    else if (subjectmarks >= 40) grade = 5;
     else grade = 0;
   }
}
    subject[j].calculategrade();
  }
}

void computeSGPA() {
    int totalcredits = 0
    int totalpoints = 0;

for (int j = 0; j < 8; j++) {
    total credits += subject[j].credits;
    totalpoints += subject[j].grade * subject[j].credits;

SGPA = Total points / totalcredits;

void displayresult() {
    System.out.println("\n Student details:");
    System.out.println("Name : "+name);
    System.out.println("USN : " + usn);
for (int j = 0 ; j < 9 ; j++) {
    System.out.println("Subject "+(j+1) +
"Marks:" subject[j].subjectmarks +", Credits:"
+ Subject[j] credits +", grade:" + subjed[j].grade);
```

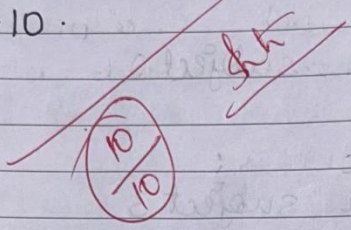
```
    }
    System.out.printf("SGPA : %.2f \n", SGPA);
}

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

Student[] students = new Student[numstudents];

for(int i=0; i<numstudents; i++){
System.out.println("Enter the details for Student
    (i+1)+" :");
Students[i]= new student();
students[i].getStudentDetails();
students[i].getmarks();
students[i].computeSGPA();
students[i].displayresult();
}
sc.close();

}
}
```

Output:

Enter the number of Students : 2.

Enter the name = meghana.
Enter the USN = 24BECS401
Enter marks : 23.
Enter credits : 2
Enter details for subject 2
Enter marks : 34
Enter credits : 5
Enter details for subject 3
Enter marks : 52
Enter credits : 6
Enter the details for subject 4
Enter marks : 49
Enter credits : 4
Enter the details for Subject 5
Enter marks : 58
Enter credits : 5
Enter details for Subject 6
Enter marks : 69
Enter credits : 6
Enter details for subject 7
Enter marks : 72
Enter credits : 5
Enter details for subject 8
Enter marks : 84
Enter credits : 4.

Subject 1 marks : 23, Credits 2, grade 0
Subject 2 marks 34, Credits 5, grade 0
Subject 3 marks 45, Credits 3, grade 1
Subject 4 marks 67, Credits 4, grade 3
Subject 5 marks 67, Credits 3, grade 3
Subject 6 marks 56, Credits 4, grade 2
Subject 7 marks 67, Credits 5, grade 3
Subject 8 marks 78, Credits 4, grade 4
SGPA : 2.10.



Source Code:

```
import java.util.Scanner;


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


void calculateGrade() {
 if (subjectMarks >= 90) grade = 6;
 else if (subjectMarks >= 80) grade = 5;
 else if (subjectMarks >= 70) grade = 4;
 else if (subjectMarks >= 60) grade = 3;
 else if (subjectMarks >= 50) grade = 2;
 else if (subjectMarks >= 40) grade = 1;
 else grade = 0;
    }
}


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 Student USN: ");
 usn = s.nextLine();
    }


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


 if (subject[j].subjectMarks < 0 || subject[j].subjectMarks > 100) {
 System.out.println("Invalid marks! Please enter marks between 0 and 100.");
 j--;
 continue;
           }

 System.out.print("Enter credits: ");
 subject[j].credits = s.nextInt();

 if (subject[j].credits < 1 || subject[j].credits > 10) {
 System.out.println("Invalid credits! Please enter credits between 1 and 10.");
  j--;
               continue;
           }


subject[j].calculateGrade();
        }
    }
```

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

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


SGPA = weightedGradePoints / totalCredits;
    }

void displayResult() {
System.out.println("\nStudent Details:");
System.out.println("Name: " + name);
System.out.println("USN: " + usn);
for (int j = 0; j < 8; j++) {
System.out.println("Subject " + (j + 1) + " Marks: " + subject[j].subjectMarks + ", Credits: " + subject[j].credits + ", Grade: " + subject[j].grade);
        }
System.out.printf("SGPA: %.2f\n", SGPA);
    }


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

Student[] students = new Student[numStudents];
```

```
for (int i = 0; i < numStudents; i++) {
System.out.println("\nEnter details for Student " + (i + 1) + ":");
students[i] = new Student(); // Create a new Student object
students[i].getStudentDetails();
students[i].getMarks();
students[i].computeSGPA();
students[i].displayResult();
        }

sc.close();
    }
}
```

Output:

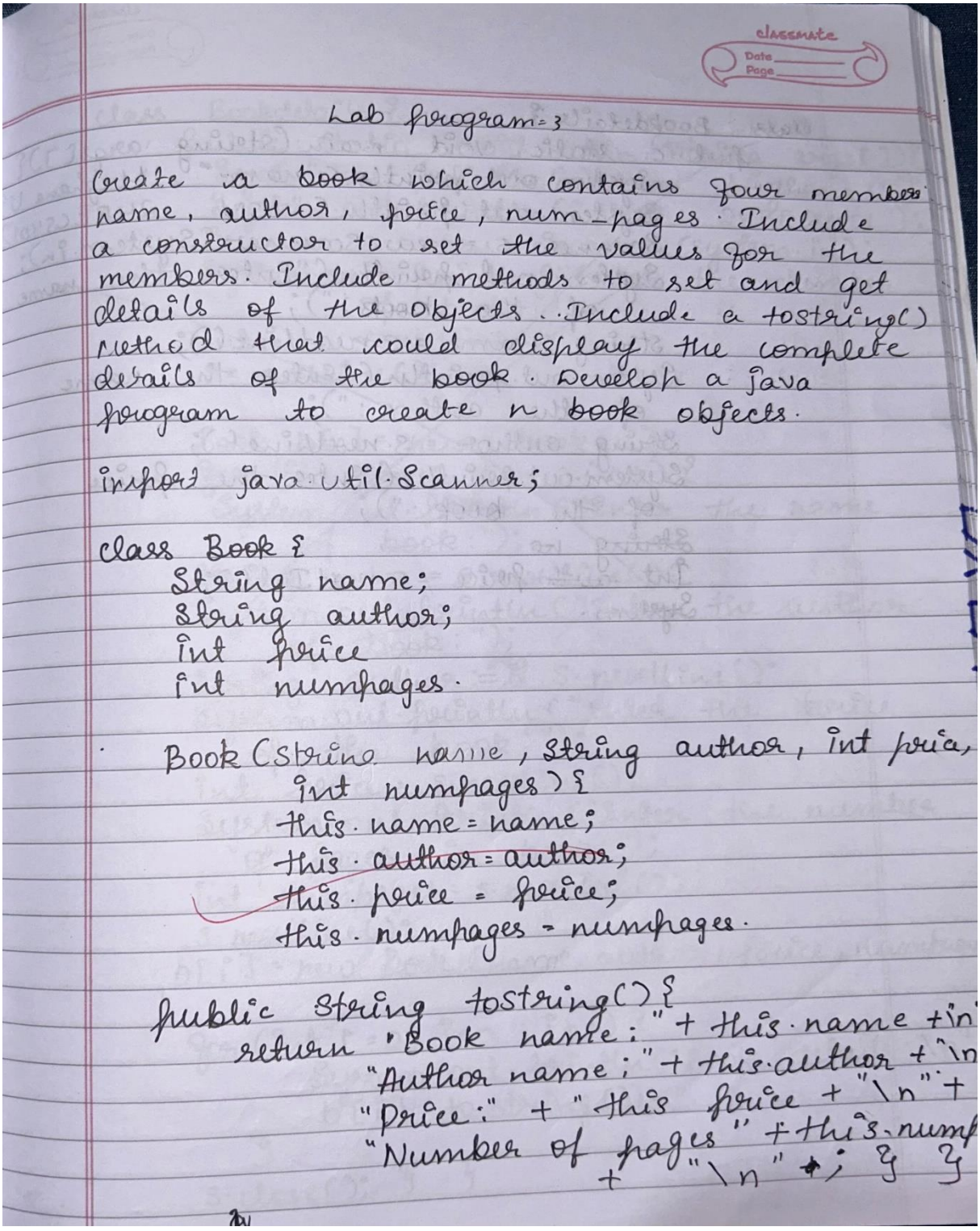
```
Enter details for Student 2:
Enter Student Name: abcd
Enter Student USN: 24becs123
Enter details for subject 1:
Enter marks: 45
Enter credits: 6
Enter details for subject 2:
Enter marks: 78
Enter credits: 9
Enter details for subject 3:
Enter marks: 78
Enter credits: 9
Enter details for subject 4:
Enter marks: 78
Enter credits: 9
Enter details for subject 5:
Enter marks: 67
Enter credits: 8
Enter details for subject 6:
Enter marks: 56
Enter credits: 9
Enter details for subject 7:
Enter marks: 67
Enter credits: 8
Enter details for subject 8:
Enter marks: 78
Enter credits: 6

Student Details:
Name: abcd
USN: 24becs123
Subject 1 Marks: 45, Credits: 6, Grade: 1
Subject 2 Marks: 78, Credits: 9, Grade: 4
Subject 3 Marks: 78, Credits: 9, Grade: 4
Subject 4 Marks: 78, Credits: 9, Grade: 4
Subject 5 Marks: 67, Credits: 8, Grade: 3
Subject 6 Marks: 56, Credits: 9, Grade: 2
Subject 7 Marks: 67, Credits: 8, Grade: 3
Subject 8 Marks: 78, Credits: 6, Grade: 4
SGPA: 3.19
```

```
D:\practice>java Student
Enter the number of students: 2

Enter details for Student 1:
Enter Student Name: meghana
Enter Student USN: 24becs401
Enter details for subject 1:
Enter marks: 23
Enter credits: 7
Enter details for subject 2:
Enter marks: 89
Enter credits: 9
Enter details for subject 3:
Enter marks: 67
Enter credits: 8
Enter details for subject 4:
Enter marks: 56
Enter credits: 8
Enter details for subject 5:
Enter marks: 45
Enter credits: 7
Enter details for subject 6:
Enter marks: 67
Enter credits: 8
Enter details for subject 7:
Enter marks: 76
Enter credits: 7
Enter details for subject 8:
Enter marks: 89
Enter credits: 8

Student Details:
Name: meghana
USN: 24becs401
Subject 1 Marks: 23, Credits: 7, Grade: 0
Subject 2 Marks: 89, Credits: 9, Grade: 5
Subject 3 Marks: 67, Credits: 8, Grade: 3
Subject 4 Marks: 56, Credits: 8, Grade: 2
Subject 5 Marks: 45, Credits: 7, Grade: 1
Subject 6 Marks: 67, Credits: 8, Grade: 3
Subject 7 Marks: 76, Credits: 7, Grade: 4
Subject 8 Marks: 89, Credits: 8, Grade: 5
SGPA: 2.97
```

PROGRAM 3:

ToString program:

Algorithm:



Lab program-3

Create a 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 details of the objects. Include a tostring() method that could display the complete details of the book. Develop a java program to create n book objects.

```
import java.util.Scanner;

class Book {
    String name;
    String author;
    int price
    int numpages.

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

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

```
class Bookdetails {
    public static void main (String args[]){
      System.out.println("Name Meghana.U");
      System.out.println("USN 24BECS401");
      Scanner s = new Scanner(System.in);
      System.out.println("Enter the number
                of books");
      int n = s.nextInt();
      s = nextLine();

      Book b[] = new book[n];
      for (int i=0; i<n ; i++) {
       System.out.println("Enter the name
            of book:");
      String name = s.nextLine();
      System.out.println("Enter the author
            of book:");
      String author = s.nextLine();
      System.out.println("Enter the price
            of the book:");
      int price = s.nextInt();
      System.out.println("Enter the number
            of pages in book:");
      int numpages = s.nextInt();
      s.nextLine();
     b[i] = new Book (name, author, price, numpages);
}
     for (int i=0; i<n ; i++) {
          System.out.println("Book details:\n"+
            b[i].toString());
        }
     s.close(); } }
```

Output:

Enter the number of book
2.

Enter the name of book
Java the complete refrence
Enter the author of book.
herbert schildt
Enter the price of book
~~25t~~ 400
Enter the number of pages in book
250.

Enter the name of book
DBMS
Enter the author of book
Rajv copra
Enter the price of book
500
Enter the number of pages in book
300

Book details
Book Name: Java the complete refrence
Author: Herber schildt
Price: 400
Pages: 250.

Book name: DBMS
Author: Rajv chopra.
Price: 500
Pages: 300.

10/10

Source Code:

```
import java.util.Scanner;

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

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

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

class Bookdetails {
    public static void main(String args[]) {
        System.out.println("name : meghana u");
        System.out.println("usn : 24BECS401");
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the number of books");
        int n = s.nextInt();
        s.nextLine();

        Book b[] = new Book[n];
        for (int i = 0; i < n; i++) {
            System.out.println("Enter the name of the book:");
            String name = s.nextLine();
            System.out.println("Enter the author of the book:");
            String author = s.nextLine();
            System.out.println("Enter the price of the book:");
            int price = s.nextInt();
            System.out.println("Enter the number of pages of the book:");
            int numpages = s.nextInt();
            s.nextLine();

            b[i] = new Book(name, author, price, numpages);
        }

        for (int i = 0; i < n; i++) {
            System.out.println("Book details are:\n" + b[i].toString());
        }
        s.close();     }
}
```

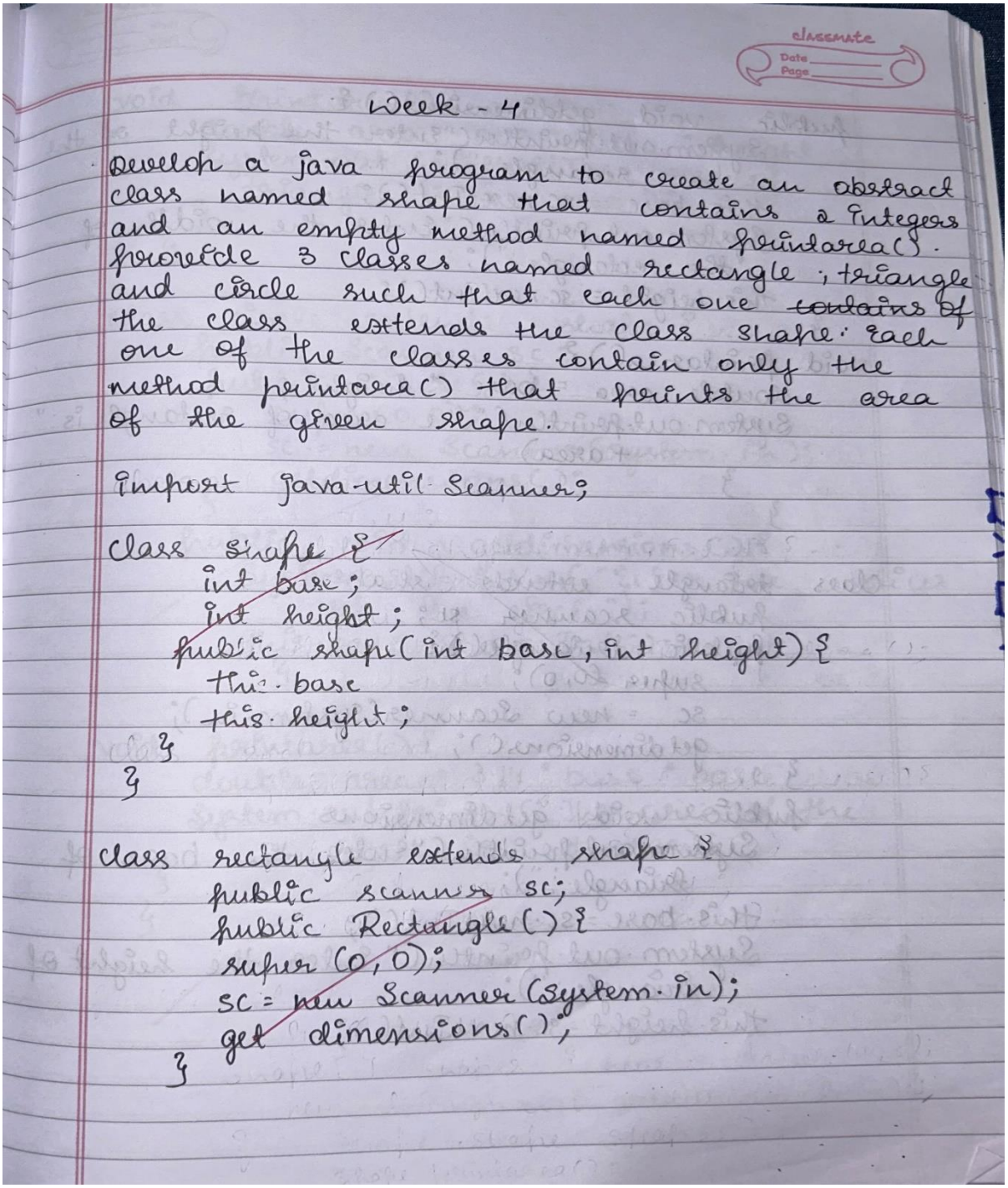
Output:

```
D:\24BECS401>java Bookdetails.java
name : meghana u
usn : 24BECS401
Enter the number of books
2
Enter the name of the book:
java the complete refrence
Enter the author of the book:
herbert schildt
Enter the price of the book:
400
Enter the number of pages of the book:
250
Enter the name of the book:
dbms
Enter the author of the book:
rajiv chopra
Enter the price of the book:
500
Enter the number of pages of the book:
300
Book details are:
Book name: java the complete refrence
Author name: herbert schildt
Price: 400
Number of pages: 250

Book details are:
Book name: dbms
Author name: rajiv chopra
Price: 500
Number of pages: 300
```

Program 4:

Area of shape program

Algorithm



Week - 4

Develop a java program to create an abstract class named shape that contains a integers and an empty method named printarea(). Provide 3 classes named rectangle, triangle and circle such that each one ~~contains~~ of the class 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;

class Shape {
    int base;
    int height;
    public shape(int base, int height) {
        this.base
        this.height;
    }
}

class rectangle extends shape {
    public scanner sc;
    public Rectangle() {
    super(0,0);
    sc = new Scanner(System.in);
    get dimensions();
}
```

```
public void getdimensions() {
    System.out.println("Enter the height of the
        rectangle:");
    this.base = sc.nextInt();
    System.out.println("Enter the width of
        the rectangle:");
    this.height = sc.nextInt();
}
void printarea() {
    double area = base * height;
    System.out.println("The area of rectangle is:"
        + area)
    }
}

class triangle extends shape {
    public scanner sc;
    public triangle() {
        super(0, 0);
        sc = new Scanner(System.in);
        getdimensions();
    }
    public void getdimensions
        System.out.println("Enter the base of
            triangle:");
        this.base = sc.nextInt();
        System.out.println("Enter the height of
            triangle:");
        this.height = sc.nextInt();
```

```
void printarea(){
    double area = 0.5 * base * height;
    System.out.println("The area of
      triangle is:" + area);
  }
}

class circle extends shape {
    public Scanner sc;
    public circle(){
        super(0, 0);
        sc = new Scanner(system.in);
        getdimensions();
    }
    public void getdimensions(){
        System.out.println("Enter the radius
            of the circle");
        this.base = sc.nextInt();
    }

void printarea(){
    double area = 3.14 * base * base;
    System.out.println("The area of the
            circle is:" + area);
  }
}
```

```
public class shapearea{
    public static void main(String args[]){
    Rectangle rectangle = new Rectangle();
    rectangle.printarea();
    Triangle triangle = new Triangle();
    triangle.printarea();
    Circle circle = new Circle();
    circle.printarea();
    }
}.
```

Output:

```
Enter the height of rectangle: 23
Enter the bwidth of rectangle: 34.
The area of rectangle : 782.0

Enter the base of triangle: 4.
Enter the height of triangle: 3
The area of ~~rectc~~ triangle : 6.0

Enter the radius of Circle: 3.
The area of the circle is: 28.2599
```

10/10 21/10/24

Source Code:

```
import java.util.Scanner;

class Shape {
     int base;
     int height;

    public Shape(int base, int height) {
        this.base = base;
        this.height = height;
    }
}

class Rectangle extends Shape {
    public Scanner sc;

    public Rectangle() {
        super(0, 0);
        sc = new Scanner(System.in);
        getDimensions();
    }

    public void getDimensions() {
        System.out.print("Enter the height of the rectangle: ");
        this.base = sc.nextInt();
        System.out.print("Enter the width of the rectangle: ");
        this.height = sc.nextInt();
    }

    void printArea() {
        double area =  base * height;
        System.out.println("The area of the rectangle is: " + area);
    }
}
class Triangle extends Shape {
    public Scanner sc;

    public Triangle() {
        super(0, 0);
        sc = new Scanner(System.in);
        getDimensions();
    }

    public void getDimensions() {
        System.out.print("Enter the base of the triangle: ");
        this.base = sc.nextInt();
        System.out.print("Enter the height of the triangle: ");
        this.height = sc.nextInt();
    }
```

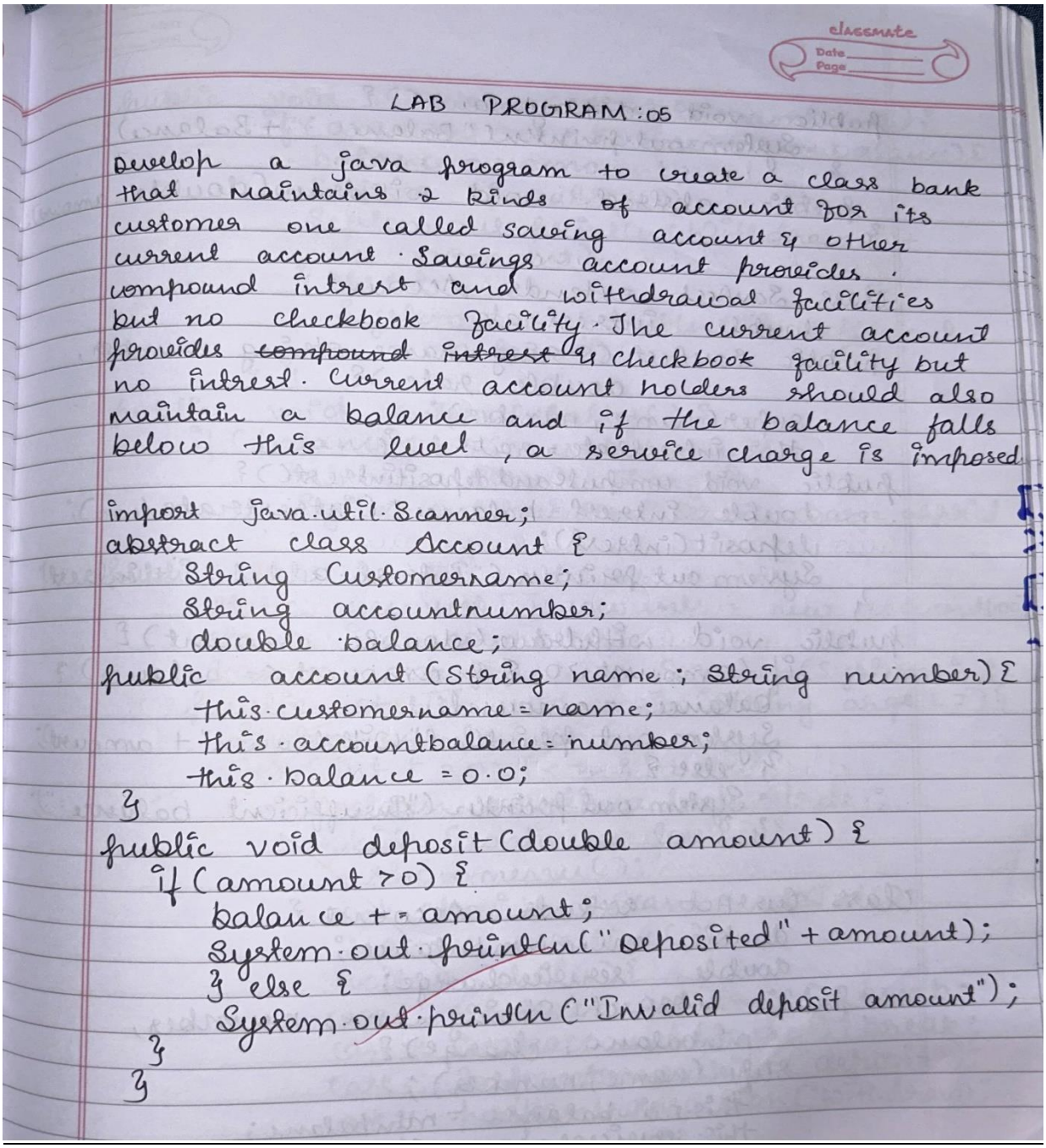
```
    void printArea() {
        double area = 0.5 * base * height;
        System.out.println("The area of the triangle is: " + area);
    }
}

class Circle extends Shape {
    public Scanner sc;

    public Circle() {
        super(0,0);
        sc = new Scanner(System.in);
        getDimensions();
    }

    public void getDimensions() {
        System.out.print("Enter the radius of the Circle: ");
        this.base = sc.nextInt();
            }

    void printArea() {
        double area = 3.14 * base * base;
        System.out.println("The area of the circle is: " + area);
    }
}


 public class shapearea{
    public static void main(String[] args) {
        Rectangle rectangle = new Rectangle();
        rectangle.printArea();
        Triangle triangle = new Triangle();
        triangle.printArea();
        Circle circle = new Circle();
        circle.printArea();


    }
}
```

Output:

```
PS D:\24BECS401> java shapearea
Enter the height of the rectangle: 23
Enter the width of the rectangle: 34
The area of the rectangle is: 782.0
Enter the base of the triangle: 4
Enter the height of the triangle: 3
The area of the triangle is: 6.0
Enter the radius of the Circle: 3
The area of the circle is: 28.259999999999998
```

Program 5:

Bank program:
Algorithm



LAB PROGRAM :05

Develop a java program to create a class bank that maintains 2 kinds of account for its customer one called saving account & other current account. Savings account provides compound intrest and withdrawal facilities but no checkbook facility. The current account provides ~~compound intrest~~ & checkbook facility but no intrest. Current account holders should also maintain a balance and if the balance falls below this level, a service charge is imposed.

```
import java.util.Scanner;
abstract class Account {
    String Customername;
    String accountnumber;
    double balance;
public account (String name ; String number) {
    this.customername = name;
    this.accountbalance = number;
    this.balance = 0.0;
}
public void deposit (double amount) {
  if (amount > 0) {
      balance += amount;
      System.out.println("Deposited" + amount);
      } else {
       System.out.println("Invalid deposit amount");
  }
}
```

```
public void displaybalance() {
    System.out.println("Balance :" + Balance);
}
public abstract void withdraw(double amount);
}

class SavAcct extends Account
    double interestrate
public SavAcct (String name, string number,
                double rate ){
    super( name, number);
    this.interestrate = rate; }
public void computeanddepositintrest() {
    double intrest = balance * (interestrate/100);
    deposit(intrest);
    System.out.println ("Intrest added :" + intrest);
}
public void withdraw (double amount) {
    if (amount >0 && amount <= balance) {
        balance -= amount;
        System.out.println("withdrawn" + amount);
    } else {
        System.out.println ("Insufficient balance:");
    } }

class CurrAct extends Account {
    double minumbalance;
    double servicecharge;
public currAct (String name, number,
        minbalance, charge) {
    super(name, number);
    this.minbalance = minbalance;
    this.servicecharge = charge;
```

```
public void withdraw(double amount){
    if(amount >=0 && amount <= balance){
        balance -= amount;
        impose penalty();
        System.out.println("Withdrawn" +
            amount);
    } else {
        System.out.println("Insufficient
            balance");
    }}
public void impose penalty(){
    if(balance < minimumbalance){
        balance -= service charge;
        System.out.println("Service charge applied
public class Bank {
    static account[] accounts = new account[100]
    static int count = 0;
    Scanner scanner = new Scanner(System.in);
    public static void main(String args[]){
        System.out.println
        for(int i = 0; i<n; i++){
            boolean continue loop = true;
            while(continue loop){
                showmenu();
                int choice = scanner.nextInt();
                scanner.nextline();
                switch(choice){
                case 1: createaccount(true); break;
                case 2: createaccount(false); break;
                case 3: performtransaction("deposit");
                case 4: performtransaction("withdraw");
```

```
        case5 : computeInterest();
        case6 : displaybalance(); break.
        default System.out.println("Invalid choic
    } scanner.close(); }

public static void showmenu(){
    S.O ("create savings account:)
        ("Create current account
        ("Deposit
        ("withdraw");
        ("compute interest balance");
        ("display balance");
        (" Exit");

public static void create account(boolean issavings
    if (count < accounts.length) {
        System.out.println("name")
        String name = Scanner.nextLine();
        System.out.println("account number");
        int number = Scanner.nextInt();

    if(issavings) {
        System.out.print("Enter interest rate:")
        double rate = scanner.nextdouble();
        accounts[count++] = new Savacct(name,
                number, rate);
    } else {
        System.out.println("Enter minimum
                balance")
        double minbalance = scanner.nextdouble()
        System.out.println("Enter service charg
        double charge = scanner.nextdouble();
```

```
accounts[count++] = new CreAcct(name, number,
            minbalance, charge);
    }
    Scanner.nextLine();

public static void performtransaction(String)
    System.out.print(account number)
    System.out.print("Enter amount)

    for(Account account : accounts)
        if (type.equals("deposit"));
            account.deposit(amount");
        } else {
            account withdraw(amount);
        }
        return
} }

public static void computeinterest() {
    System.out.println("Enter account number");
    String number = scanner.nextline();
    for(Acount account : accounts){
        if (account istance of SaveAcct && Account.
            accountnumber.equals(number)) {
            ((SaveAcct) account).computeanddepositinterest;
            return; } }

public static void displaybalance() {
    System.out.println("Enter account number:");
    for(Account account : accounts) {
        account.displaybalance();
    } }
} }
```

Output:
Create saving account
Create current account
Deposit
Withdraw
Compute intrest
Banlance.
Choose an option 1
Enter customer name: meghana.
Enter account number: 12345
Enter intrest rate: 4.
Saving account created.

Create saving account
Create current account
deposit
withdraw
Computeintrest
Balance
Choose an option 3
Enter account number: 12345
Enter amount: 400

Create saving account
Create current account
deposit
withdraw
Computeintrest
Balance
choose an option
Enter account number: 12345
Deposited: 16.0

Create savings account
Create current account
Deposit
Withdraw
Compute interest
Display Balance
Choose an option 6
Enter account number 12345
Current balance: 416.0

10/10

29/10/24

Source Code:

```
import java.util.Scanner;

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

    public Account(String name, String number) {
        this.customerName = name;
        this.accountNumber = number;
        this.balance = 0.0;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

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

    public abstract void withdraw(double amount);
}

class SavAcct extends Account {
    double interestRate;

    public SavAcct(String name, String number, double rate) {
        super(name, number);
        this.interestRate = rate;
    }

    public void computeAndDepositInterest() {
        double interest = balance * (interestRate / 100);
        deposit(interest);
        System.out.println("Interest added: " + interest);
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
```

```
class CurAcct extends Account {
    private final double minimumBalance;
    private final double serviceCharge;

    public CurAcct(String name, String number, double minBalance, double charge) {
        super(name, number);
        this.minimumBalance = minBalance;
        this.serviceCharge = charge;
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            imposePenalty();
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance.");
        }
    }

    public void imposePenalty() {
        if (balance < minimumBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge applied: " + serviceCharge);
        }
    }
}

public class Bank {
    static Account[] accounts = new Account[100];
    static int count = 0;
    static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        System.out.print("Enter the number of customers: ");
        int n = scanner.nextInt();
        scanner.nextLine();

        for (int i = 0; i < n; i++) {
            boolean continueLoop = true;
            while (continueLoop) {
                showMenu();
                int choice = scanner.nextInt();
                scanner.nextLine();

                switch (choice) {
                    case 1: createAccount(true); break;
                    case 2: createAccount(false); break;
                    case 3: performTransaction("deposit"); break;
                    case 4: performTransaction("withdraw"); break;
```

```
                case 4: performTransaction("Withdraw"); break;
                case 5: computeInterest(); break;
                case 6: displayBalance(); break;
                case 7: continueLoop = false; break;
                default: System.out.println("Invalid choice. Try again.");
            }
        }
    }
    scanner.close();
}

public static void showMenu() {
    System.out.println("\n1. Create Savings Account");
    System.out.println("2. Create Current Account");
    System.out.println("3. Deposit");
    System.out.println("4. Withdraw");
    System.out.println("5. Compute Interest");
    System.out.println("6. Display Balance");
    System.out.println("7. Exit");
    System.out.print("Choose an option: ");
}

public static void createAccount(boolean isSavings) {
    if (count < accounts.length) {
        System.out.print("Enter customer name: ");
        String name = scanner.nextLine();
        System.out.print("Enter account number: ");
        String number = scanner.nextLine();

        if (isSavings) {
            System.out.print("Enter interest rate: ");
            double rate = scanner.nextDouble();
            accounts[count++] = new SavAcct(name, number, rate);
        } else {
            System.out.print("Enter minimum balance: ");
            double minBalance = scanner.nextDouble();
            System.out.print("Enter service charge: ");
            double charge = scanner.nextDouble();
            accounts[count++] = new CurAcct(name, number, minBalance, charge);
        }
        scanner.nextLine();
        System.out.println((isSavings ? "Savings" : "Current") + " Account created.");
    } else {
        System.out.println("Max accounts reached.");
    }
}

public static void performTransaction(String type) {
    System.out.print("Enter account number: ");
    String number = scanner.nextLine();
    System.out.print("Enter amount: ");
    double amount = scanner.nextDouble();
```

```
    public static void performTransaction(String type) {
        System.out.print("Enter account number: ");
        String number = scanner.nextLine();
        System.out.print("Enter amount: ");
        double amount = scanner.nextDouble();
        scanner.nextLine();

        for (Account account : accounts) {
            if (account != null && account.accountNumber.equals(number)) {
                if (type.equals("deposit")) {
                    account.deposit(amount);
                } else {
                    account.withdraw(amount);
                }
                return;
            }
        }
        System.out.println("Account not found.");
    }

    public static void computeInterest() {
        System.out.print("Enter account number: ");
        String number = scanner.nextLine();
        for (Account account : accounts) {
            if (account instanceof SavAcct && account.accountNumber.equals(number)) {
                ((SavAcct) account).computeAndDepositInterest();
                return;
            }
        }
        System.out.println("Savings Account not found or not applicable.");
    }

    public static void displayBalance() {
        System.out.print("Enter account number: ");
        String number = scanner.nextLine();
        for (Account account : accounts) {
            if (account != null && account.accountNumber.equals(number)) {
                account.displayBalance();
                return;
            }
        }
        System.out.println("Account not found.");
    }
}
```

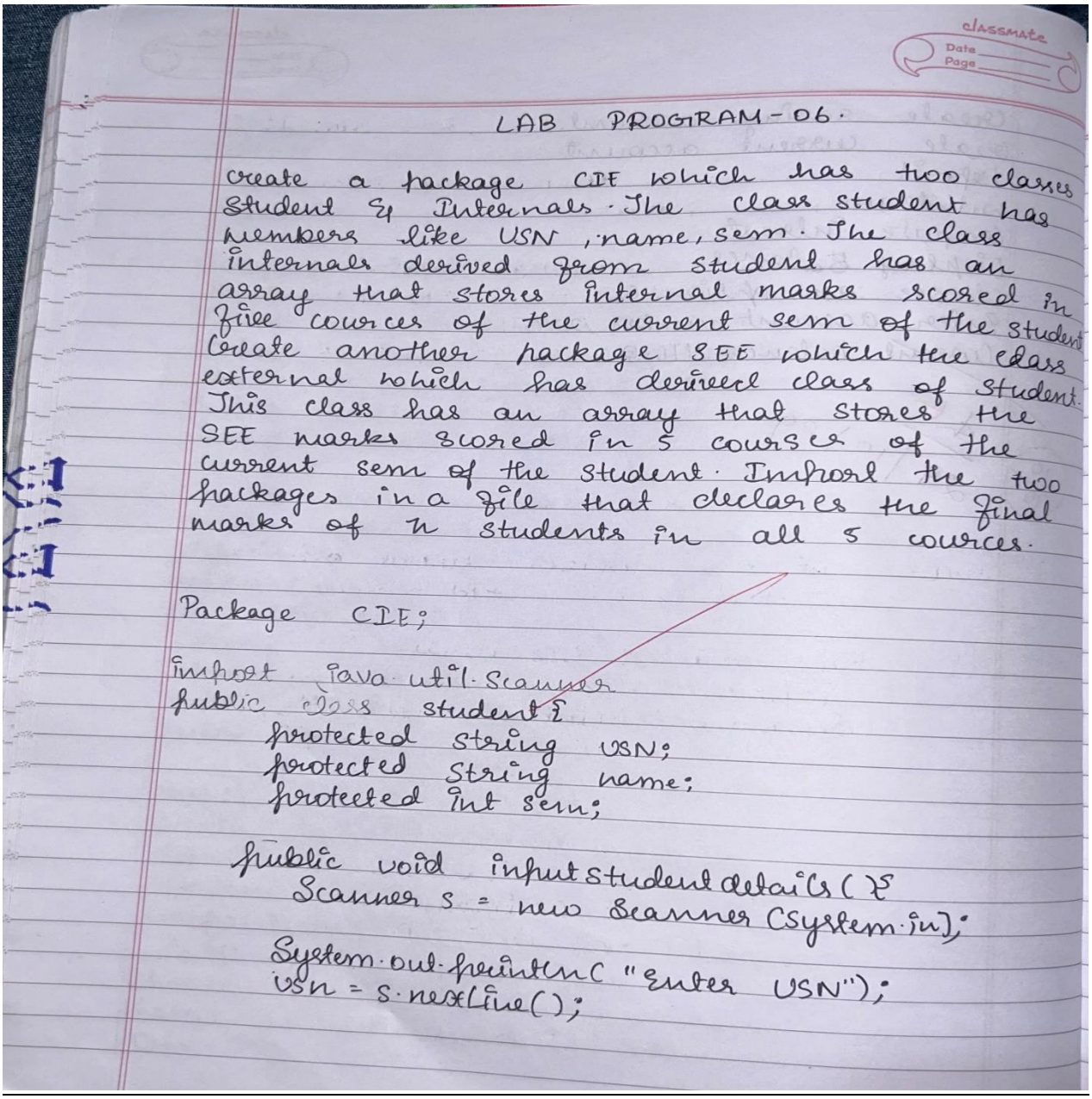
Output:

```
1. Create Savings Account
2. Create Current Account
3. Deposit
4. Withdraw
5. Compute Interest
6. Display Balance
7. Exit
Choose an option: 5
Enter account number: 12345
Deposited: 16.0
Interest added: 16.0

1. Create Savings Account
2. Create Current Account
3. Deposit
4. Withdraw
5. Compute Interest
6. Display Balance
7. Exit
Choose an option: 6
Enter account number: 12345
Current Balance: 416.0

1. Create Savings Account
2. Create Current Account
3. Deposit
4. Withdraw
5. Compute Interest
6. Display Balance
7. Exit
Choose an option: 2
Enter customer name: megha
Enter account number: 123
Enter minimum balance: 300
Enter service charge: 50
Current Account created.
```

PROGRAM 6:

Package program:

Algorithm:



LAB PROGRAM - 06.

create a package CIE which has two classes Student & Internals. The class student has members like USN, name, sem. The class internals derived from student has an array that stores internal marks scored in five courses of the current sem of the student. Create another package SEE which the class external which has derived class of student. This class has an array that stores the SEE marks scored in 5 courses of the current sem of the student. Inhorit the two packages in a file that declares the final marks of n students in all 5 courses.

```
Package CIE;

import java.util.Scanner
public class student {
        protected String USN;
        protected String name;
        protected int sem;

    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);

        System.out.println("Enter USN");
        USN = s.nextLine();
```

```
        System.out.println("Enter Name:");
        name = s.nextline();
        System.out.println("Enter sem:");
        sem = s.nextInt();
    }
    public void displaydetials(){
        System.out.println("USN" + usn);
        System.out.println("Name:" + name);
        System.out.println("sem" : sem);
    }}

Package CIE;

import java.util.Scanner;

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

    public void CIE marks(){
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the marks
            for subject " + (i+1) + ";");
        marks[i] = s.nextInt();
    }

public void displayCIEmarks(){
    System.out.println("CIE marks");
    for(int i = 0; i<5 ; i++){
        System.out.println("Subject + (i+1) :
            " + marks[i]);
    }
}}
```

```
package SEE

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

public class Externals extends Internals {
    protected int[] finalmarks = new Int[5];

    public Externals() {
        marks = new Int[5];
        final marks = new Int[5];
    }

public void inputSEE marks() {
    Scanner s = new Scanner (System.in);
    System.out.println("Enter the external
     marks for 5 subjects");
    for (int i = 0 ; i<5; i++) {
        System.out.println( Enter external
          marks for " + (i+1));
        ~~mark~~ marks[i] = s.nextInt();
    }
}

public void calculatefinalmarks() {
    for (int i = 0; i<5 ; i++) {
        finalmarks [i] = (int)(0.5 * marks[i] +
          0.5 * marks[i] );
```

```
public void displayFinalmarks() {
    displayStudentdetails();
    System.out.println("Final marks for
        the 5 subjects");
    for(int i=0; i<5; i++){
        System.out.println("subject "+(i+1)+
            "final marks" + finalmarks[i]);
    }
}

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

public class marks calculate {
    public static void main(String args[]){
        Externals externals = new externals();

        externals.input.Studentdetails();
        externals.inputCIEmarks();
        Externals.inputSEEmarks();
        externals.calculatefinalmarks();
        externals.displayfinalmarks();
    }
}
```

Output:

```
Name: Meghana.U
USN: 24BECS401
Sem: 3
Enter marks sub1: 45
Enter marks sub2 67
Enter marks sub3 68
Enter marks sub4 45
Enter marks sub5 78
Final marks
Enter marks sub1 34
Enter marks sub2 56
Enter marks sub3 37
Enter marks sub4 89
Enter marks sub5 90.
USN : 24BECS401
Name: Meghana
Sem: 3
final maks
Sub1: 34
sub2 56
Sub3 37
Sub4 89
Sub5 90
```

10/10

11/11/24

Source Code:

```
package CIE;

import java.util.Scanner;

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


    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);

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

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

        System.out.print("Enter Semester: ");
        sem = s.nextInt();
    }

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

```java
package CIE;

import java.util.Scanner;

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


    public void inputCIEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter CIE marks for 5 subjects:");

                for (int i = 0; i < 5; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = s.nextInt();
        }
    }


    public void displayCIEmarks() {
        System.out.println("CIE Marks:");
        for (int i = 0; i < 5; i++) {
            System.out.println("Subject " + (i + 1) + ": " + marks[i]);
        }
    }
}
```

```java
package SEE;

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

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


    public Externals() {
        marks = new int[5];
        finalMarks = new int[5];
    }


    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter External Marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Enter external marks for subject " + (i + 1) + ": ");
            marks[i] = s.nextInt();
        }
    }

       public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = (int) (0.5 * marks[i] + 0.5 * marks[i]);         }
    }

    // Method to display final marks
    public void displayFinalMarks() {
        displayStudentDetails();
        System.out.println("Final Marks for 5 subjects (50% CIE + 50% SEE):");
        for (int i = 0; i < 5; i++) {
            System.out.println("Subject " + (i + 1) + " final marks: " + finalMarks[i]);
        }
    }
}
```

```
import SEE.Externals;
import java.util.Scanner;

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

        Externals externals = new Externals();


        externals.inputStudentDetails();

               externals.inputCIEmarks();


        externals.inputSEEmarks();


        externals.calculateFinalMarks();

        externals.displayFinalMarks();
    }
}
```

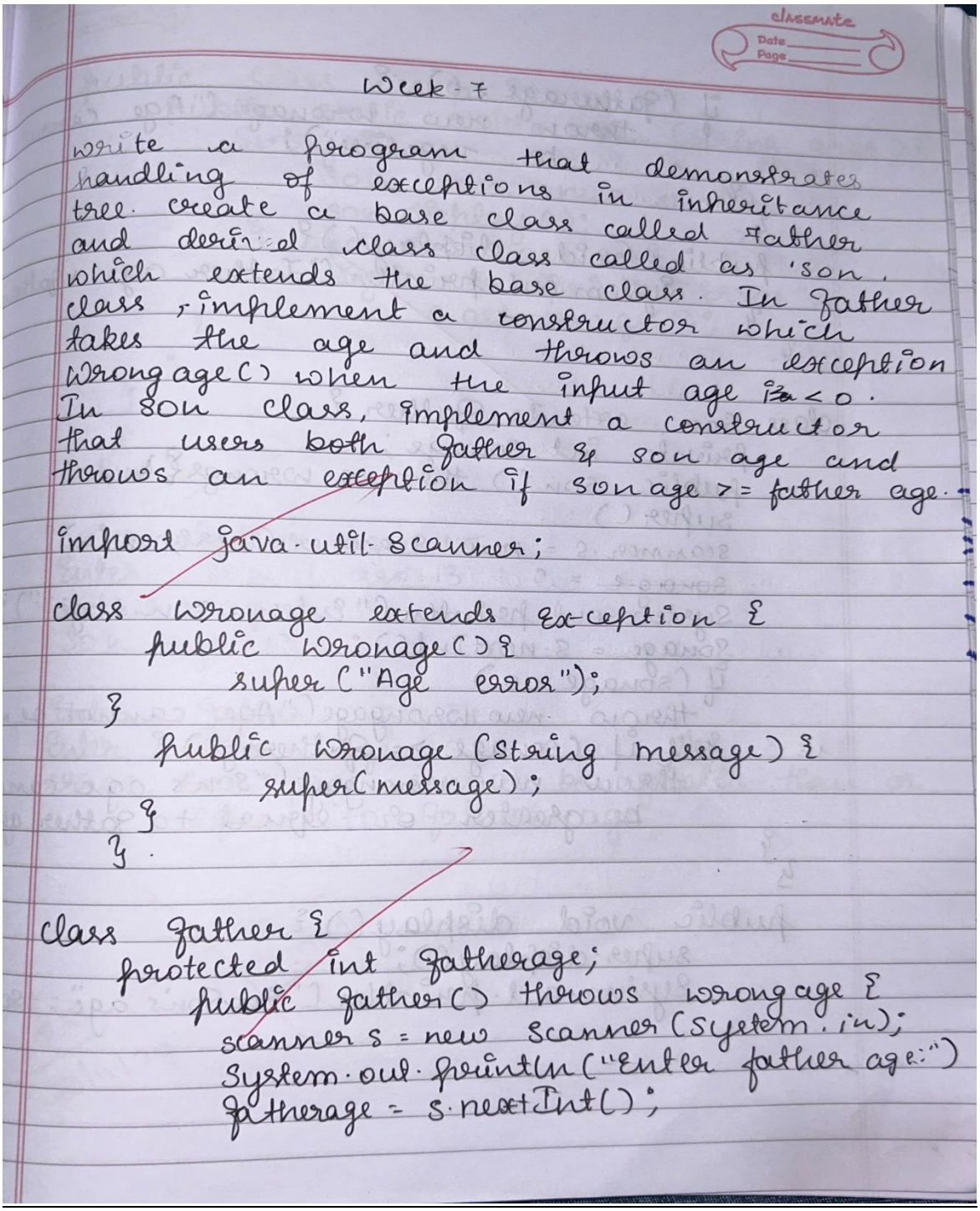
Output:

```
D:\24BECS401>java markscalculate
Enter USN: 24becs401
Enter Name: meghana
Enter Semester: 3
Enter CIE marks for 5 subjects:
Enter marks for subject 1: 45
Enter marks for subject 2: 67
Enter marks for subject 3: 68
Enter marks for subject 4: 45
Enter marks for subject 5: 78
Enter External Marks for 5 subjects:
Enter external marks for subject 1: 34
Enter external marks for subject 2: 56
Enter external marks for subject 3: 37
Enter external marks for subject 4: 89
Enter external marks for subject 5: 90
USN: 24becs401
Name: meghana
Semester: 3
Final Marks for 5 subjects (50% CIE + 50% SEE):
Subject 1 final marks: 34
Subject 2 final marks: 56
Subject 3 final marks: 37
Subject 4 final marks: 89
Subject 5 final marks: 90
```

Program 7

Exeption Handling:

Algorithm:



Week - 7

write a program that demonstrates handling of exceptions in inheritance tree. create a base class called father and derived class class called as 'son. which extends the base class. In father class, implement a constructor which takes the age and throws an exception wrongage() when the input age is < 0. In son class, implement a constructor that users both father & son age and throws an exception if son age >= father age.

```
import java.util.Scanner;

class Wronage extends Exception {
    public Wronage() {
        super("Age error");
    }
    public Wronage(String message) {
        super(message);
    }
}

class father {
    protected int fatherage;
    public father() throws wrongage {
        scanner s = new scanner(System.in);
        System.out.println("Enter father age:")
        fatherage = s.nextInt();
```

```
        if (fatherage < 0) {
            throw new wrongage("Age cannot
                be negative");
        }
    }
    public void display() {
        System.out.println("Father age: " + fatherage);
    }
}

class Son extends Father {
    private int sonage;
    public Son() throws wrongage {
        super()
        scanner s = new Scanner(System.in);
        sonage =
        System.out.println("Enter son age:");
        sonage = s.nextInt();
        if (sonage < 0)
            throw new wrongage("Age cannot be negative
        if (sonage >= fatherage) {
            throw new wrongage("son's age cannot
                be greater or equal to father age");
    }
}

public void display() {
    super.display();
    System.out.println("Son's age: " + sonage);
```

```
public class exep {
    public static void main (String args) {
        try {
            Son son = new son();
            son.display();
        } catch (wronage e) {
            System.out.println("error" +
                e.getmessage());
    }
}
}
```

Output:

Enter father age: 45
Enter son age: 13
Father age = 45
Son age = 13

Enter father age: 34
Enter son age: 56
error: Son's age cannot be greater than or equal to father's age.

10/10

28/11/24

Source Code:

```
import java.util.Scanner;

class WrongAge extends Exception {

    public WrongAge() {
        super("Age Error");
    }


    public WrongAge(String message) {
        super(message);
    }
}


class Father {
    protected int fatherAge;

       public Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter father's age: ");
        fatherAge = s.nextInt();

        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
```

```
    public void display() {
        System.out.println("Father's age: " + fatherAge);
    }
}


class Son extends Father {
    private int sonAge;


    public Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.print("Enter son's age: ");
        sonAge = s.nextInt();

        if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        } else if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to father's age");
        }
    }


    public void display() {
        super.display();
        System.out.println("Son's age: " + sonAge);
    }
}
```

```
public class exep {
    public static void main(String[] args) {
        System.out.println("Name:Meghana U");
        System.out.println("USN:24BECS401");
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Output:

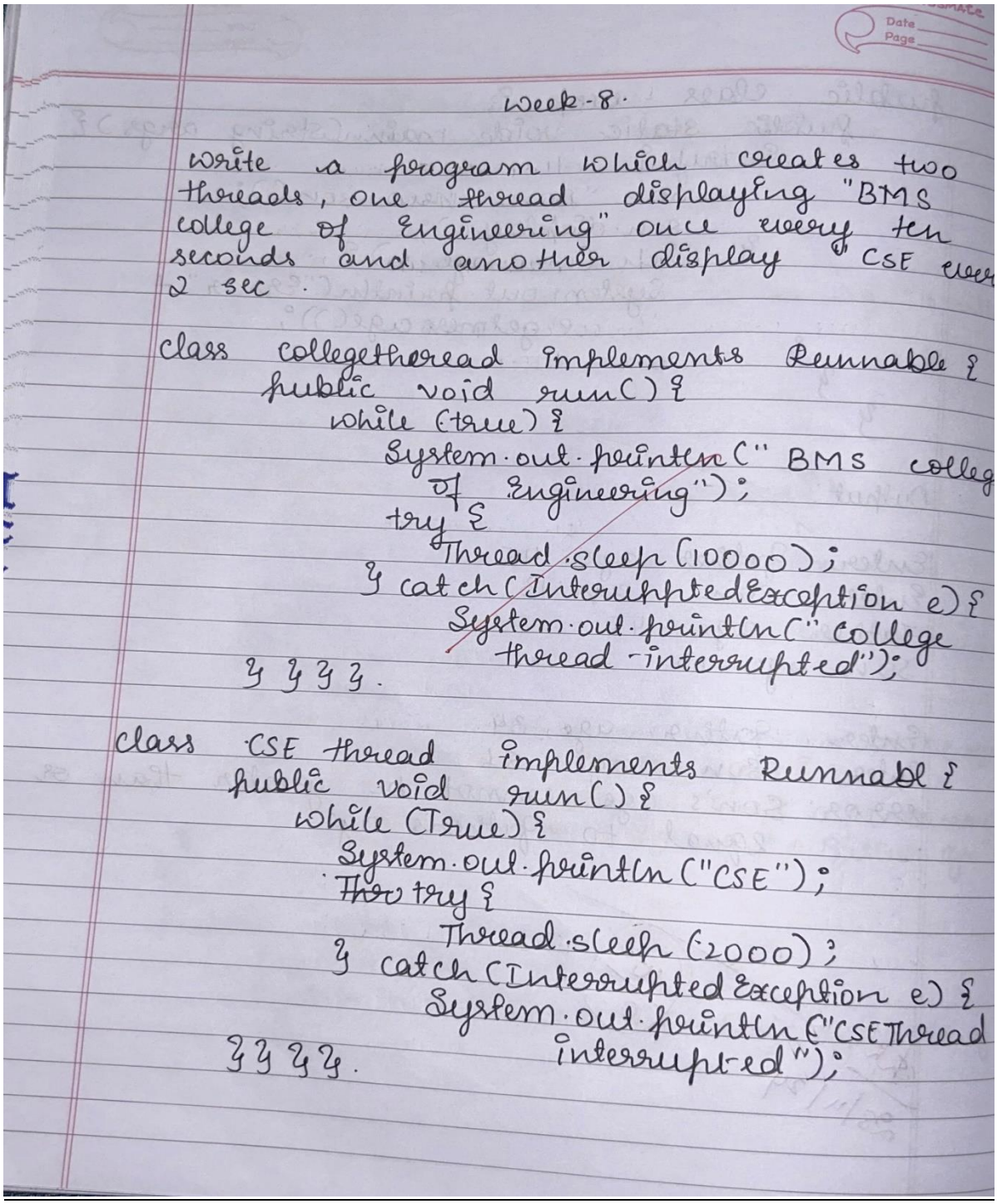
```
PS D:\practice> java exep
Name:Meghana U
USN:24BECS401
Enter father's age: 45
Enter son's age: 13
Father's age: 45
Son's age: 13
```

```
PS D:\practice> java exep
Enter father's age: 34
Enter son's age: 56
Error: Son's age cannot be greater than or equal to father's age
```

PROGRAM 8

Thread concept:

Algorithm:



Week-8.

Write a program which creates two threads, one thread displaying "BMS college of Engineering" once every ten seconds and another display "CSE ever 2 sec.

```
class collegethread implements Runnable {
    public void run() {
        while (true) {
            System.out.println(" BMS colleg
                of Engineering");
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                System.out.println(" College
                    thread -interrupted");
    } } } }.

class CSE thread implements Runnabl {
    public void run() {
        while (True) {
            System.out.println("CSE");
            try {
                Thread.sleep (2000);
            } catch (InterruptedException e) {
                System.out.println("CSE Thread
                    interrupted");
    } } } }.
```

```
public class thread1 {
    public static void main (String[] args){
        thread collegethread = new thread
                (new collegethread());
        thread csethread = new thread
                (new csethread());

        collegethread.start();
        csethread.start();
    }
}
```

output.

BMS college of Engineering.
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE

10/10

Sat
28/11/24

```
class CollegeThread implements Runnable {

    public void run() {
        while (true) {
            System.out.println("BMS College of Engineering");
            try {

                Thread.sleep(10000);
            } catch (InterruptedException e) {
                System.out.println("CollegeThread interrupted");
            }
        }
    }
}

class CSEThread implements Runnable {

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

public class thread1 {
    public static void main(String[] args) {

        Thread collegeThread = new Thread(new CollegeThread());
        Thread cseThread = new Thread(new CSEThread());

     for (int i = 0; i < 3; i++) {
        collegeThread.start();
        cseThread.start();

   }         }
}
```

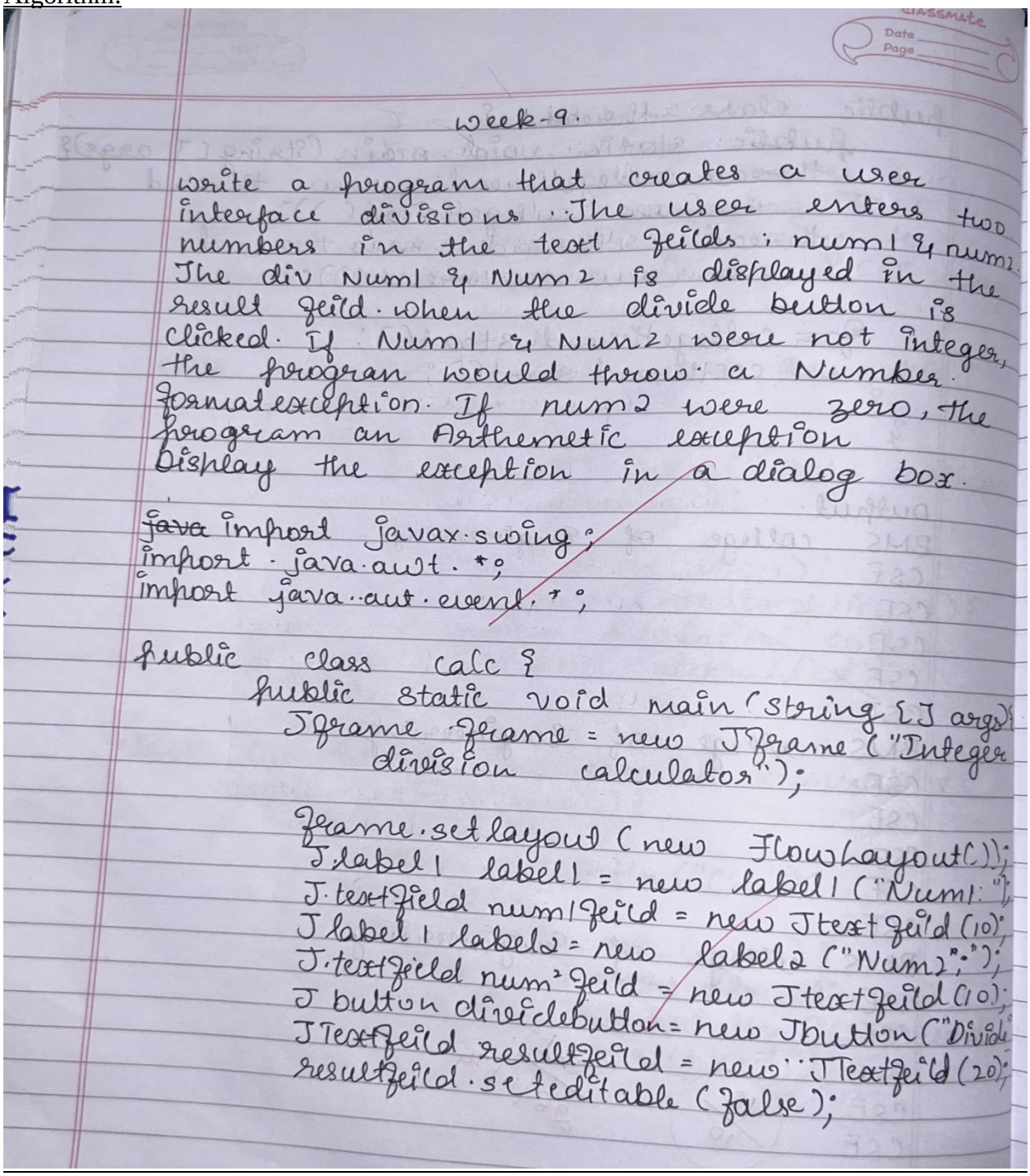
Output:

```
PS D:\practice> java thread1
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
PS D:\practice> java thread1
BMS College of Engineering
CSE
CSE
CSE
CSE
```

PROGRAM 9

Calculator:

Algorithm:



Week-9.

Write a program that creates a user interface divisions. The user enters two numbers in the text fields; num1 & num2. The div Num1 & Num2 is displayed in the result field. When the divide button is clicked. If Num1 & Num2 were not integers, the program would throw a NumberFormatException. If num2 were zero, the program an Arthemetic exception Display the exception in a dialog box.

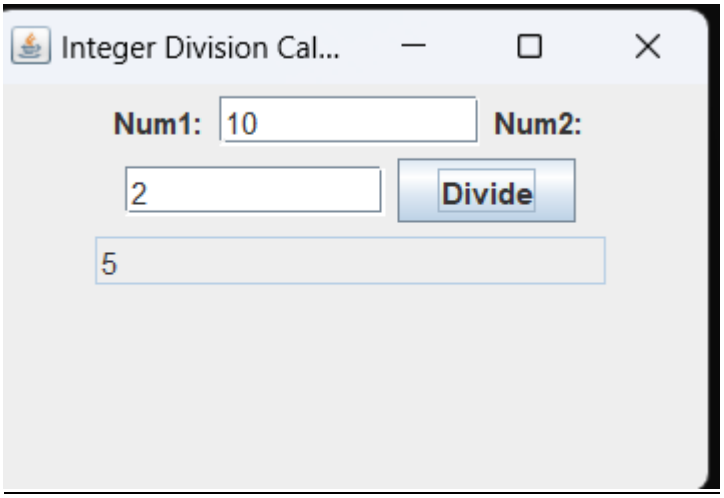
```
import javax.swing;
import java.awt.*;
import java..awt.event.*;

public class calc {
    public static void main(String[] args){
        JFrame jframe = new JFrame("Integer
                division calculator");

        jframe.setLayout(new FlowLayout());
        JLabel label1 = new label1("Num1:");
        J.textfield num1feild = new Jtextfeild(10);
        J label1 label2 = new label2("Num2:");
        J.textfield num2feild = new Jtextfeild(10);
        J button dividebutton = new JButton("Divide
        JTextfeild resultfeild = new JTextfeild(20);
        resultfeild.seteditable(false);
```

```
frame.add(label1);
frame.add(num1field);
frame.add(label2);
frame.add(num2field);
frame.add(dividebutton);
frame.add(resultfield);

frame.setsize(300,200);

frame.setDefaultclose operation
        (JFrame.EXIT_ON_CLOSE);

setframe.setvisible(true);

dividebutton.addactionlistener(new Action
        listner(){
    public void actionperformed(Action
                    Event e){
        try{
            int num1 = Integer.parseInt(num1
            feild.getText())
            int num2 = Integer.parseInt(
            num2field.getText());
            if(num2==0){
                throw new Arthematic Exception
                ("cannot divide by zero");
            }
            int result = num1/num2;
        resultfield.settext(string.valueof
            result));}
```

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

public class calc {
    public static void main(String[] args) {
                JFrame frame = new JFrame("Integer Division Calculator");


        frame.setLayout(new FlowLayout());


        JLabel label1 = new JLabel("Num1: ");
        JTextField num1Field = new JTextField(10);
        JLabel label2 = new JLabel("Num2: ");
        JTextField num2Field = new JTextField(10);
        JButton divideButton = new JButton("Divide");
        JTextField resultField = new JTextField(20);
        resultField.setEditable(false);


        frame.add(label1);
        frame.add(num1Field);
        frame.add(label2);
        frame.add(num2Field);
        frame.add(divideButton);
        frame.add(resultField);


        frame.setSize(300, 200);


        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


        frame.setVisible(true);
```

```
        frame.setVisible(true);


        divideButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {

                    int num1 = Integer.parseInt(num1Field.getText());
                    int num2 = Integer.parseInt(num2Field.getText());


                    if (num2 == 0) {

                        throw new ArithmeticException("Cannot divide by zero.");
                    }
                    int result = num1 / num2;


                    resultField.setText(String.valueOf(result));

                } catch (NumberFormatException ex) {

                    JOptionPane.showMessageDialog(frame, "Invalid input. Please enter integers only.",
                            "NumberFormatException", JOptionPane.ERROR_MESSAGE);
                } catch (ArithmeticException ex) {

                    JOptionPane.showMessageDialog(frame, ex.getMessage(),
                            "ArithmeticException", JOptionPane.ERROR_MESSAGE);
                }
            }
        });
    }
}
```

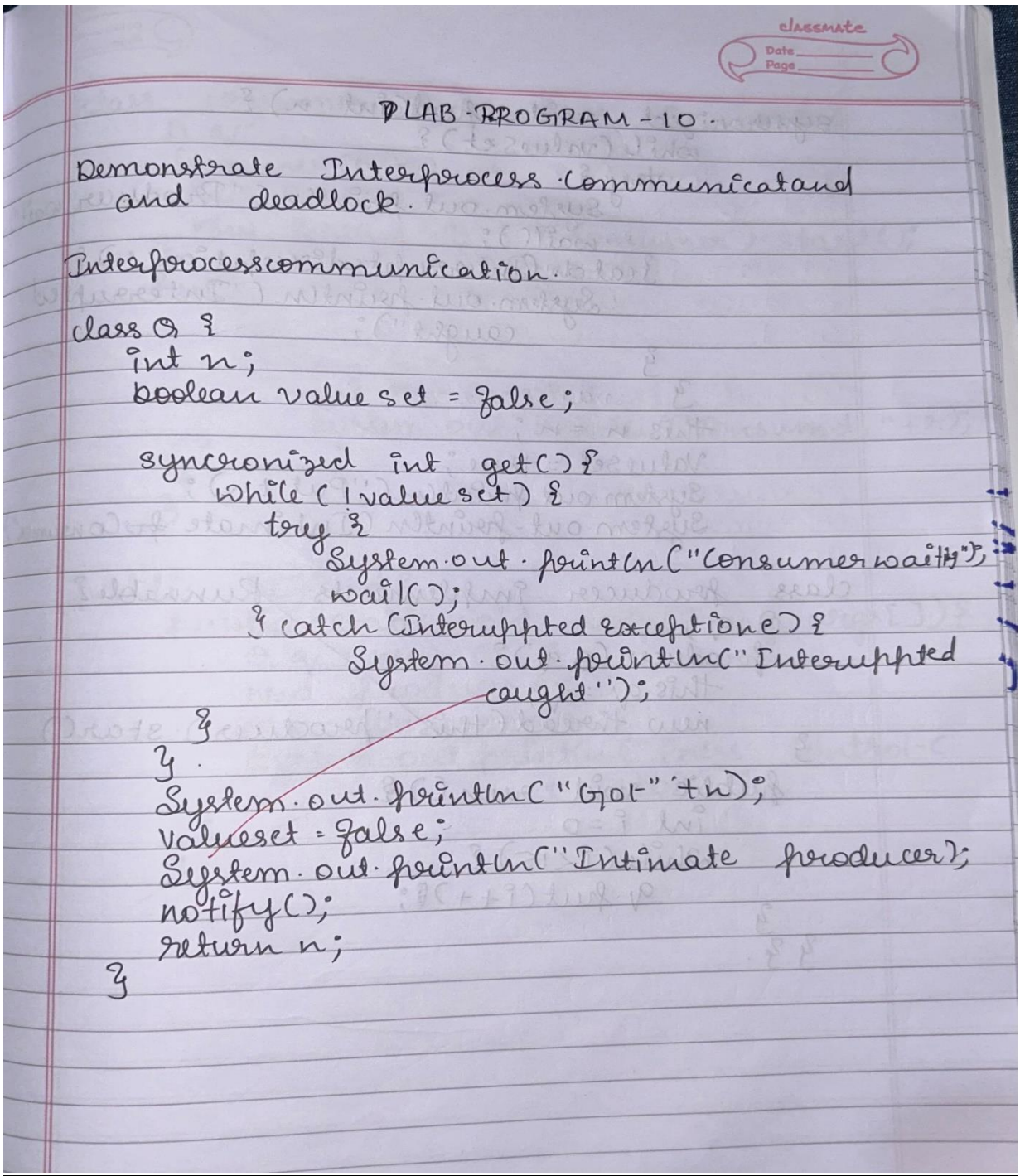
Output:



PROGRAM 10

Interprocess communication and DeadLock:

Algorithm:



PLAB-PROGRAM-10.

Demonstrate Interprocess communicatand and deadlock.

Interprocesscommunication.

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

    syncronized int get() {
        while (!valueset) {
            try {
                System.out.println("Consumer waiting");
                wait();
            } catch (Interrupted exception e) {
                System.out.println" Interrupted
                    caught");
            }
        }
        System.out.println("Got" +n);
        valueset = false;
        System.out.println("Intimate producer");
        notify();
        return n;
    }
```

```
synchronized void put(int n) {
        while (valueset) {
            try {
                System.out.println("Producer wait");
                wait();
            } catch (InterruptedException e) {
                System.out.println("Interrupted
                        caught");
            }
        }
        this.n = n;
        Valueset = true;
        System.out.println("Put " + n);
        System.out.println("Intimate to Consumer");
    }

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++);
        }
    }
}
```

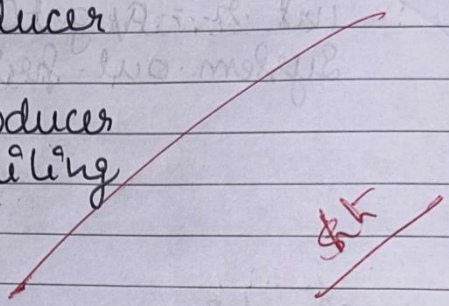
```
class consumer implements Runnable {
    Q q;
    Consumer(Q q)
    this.q = q
    new thread (this, "Consumer").start();
}
public void run(){
    int i=0;
    while (i<15) {
        int r = q.get();
        System.out.println("Consumed, "+r);
    }
}
}

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

Output:

press control c to stop
Put 0

Intimate consumer
Producer consumer
Got 1.

Intimate producer
put : 1
Intimate producer
producer wailing
Consumer : 0
got : 1



Deadlock

```
class A {
    Syncronized 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 {
    Syncronized 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 t
          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("Mai
                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:
RacingThread entered B.bar.
MainThread entered A.Foo
MainThread trying to call B.last()
Inside B.last
Back in main Thread
Racingthread trying to call A.last()
Inside A.last
Back in other Thread.

2/12/24

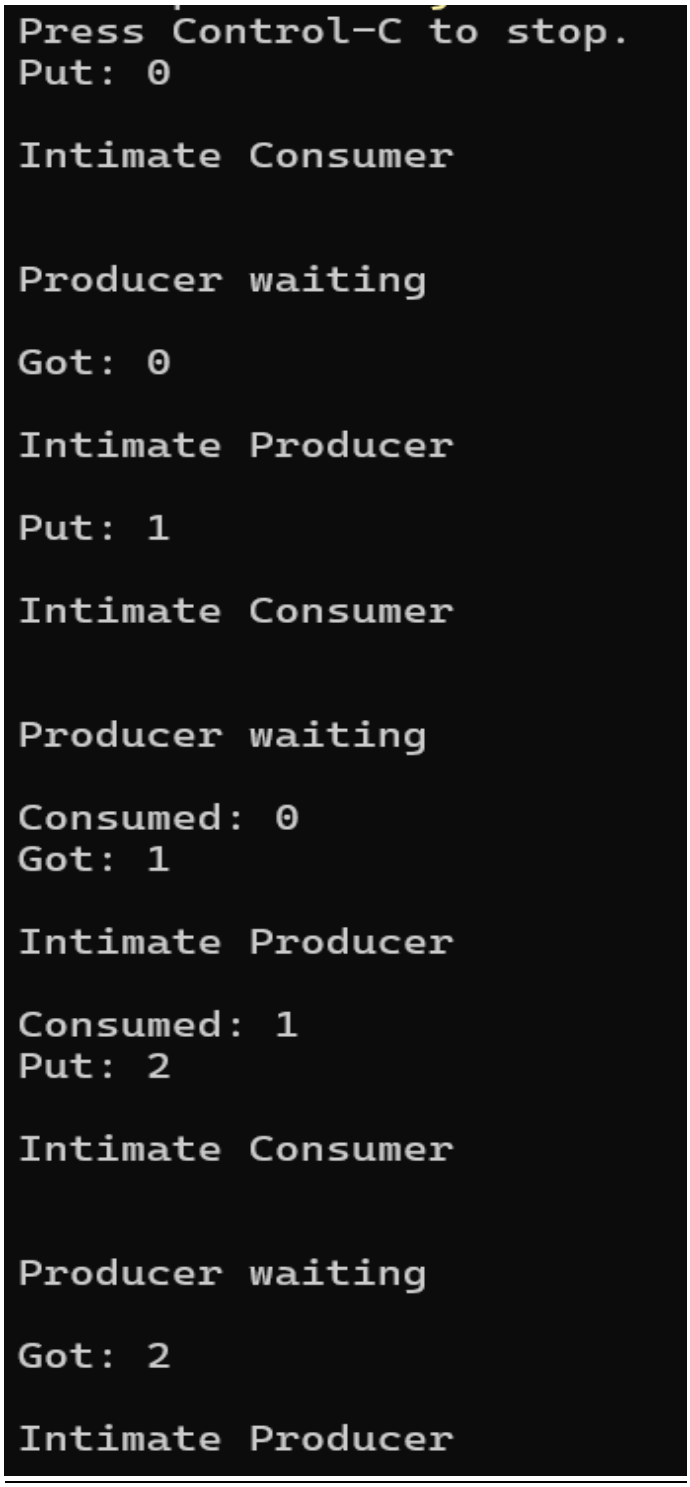
10/10

Source 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++;
        }
    }
}

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

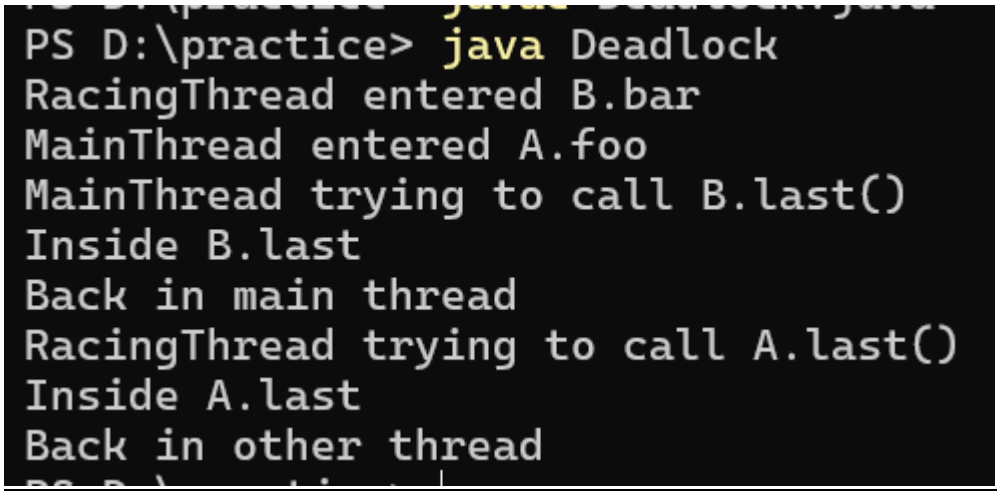
Output:


Press Control-C to stop.
Put: 0
Intimate Consumer
Producer waiting
Got: 0
Intimate Producer
Put: 1
Intimate Consumer
Producer waiting
Consumed: 0
Got: 1
Intimate Producer
Consumed: 1
Put: 2
Intimate Consumer
Producer waiting
Got: 2
Intimate Producer

```
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);
        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:


PS D:\practice> java Deadlock
RacingThread entered B.bar
MainThread entered A.foo
MainThread trying to call B.last()
Inside B.last
Back in main thread
RacingThread trying to call A.last()
Inside A.last
Back in other thread