VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Aditya Sharma (1BM22CS021)

in partial fulfilment 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

<u>CERTIFICATE</u>

This is to certify that the Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by **Aditya Sharma (1BM23CS021),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment 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. Prasad G R Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

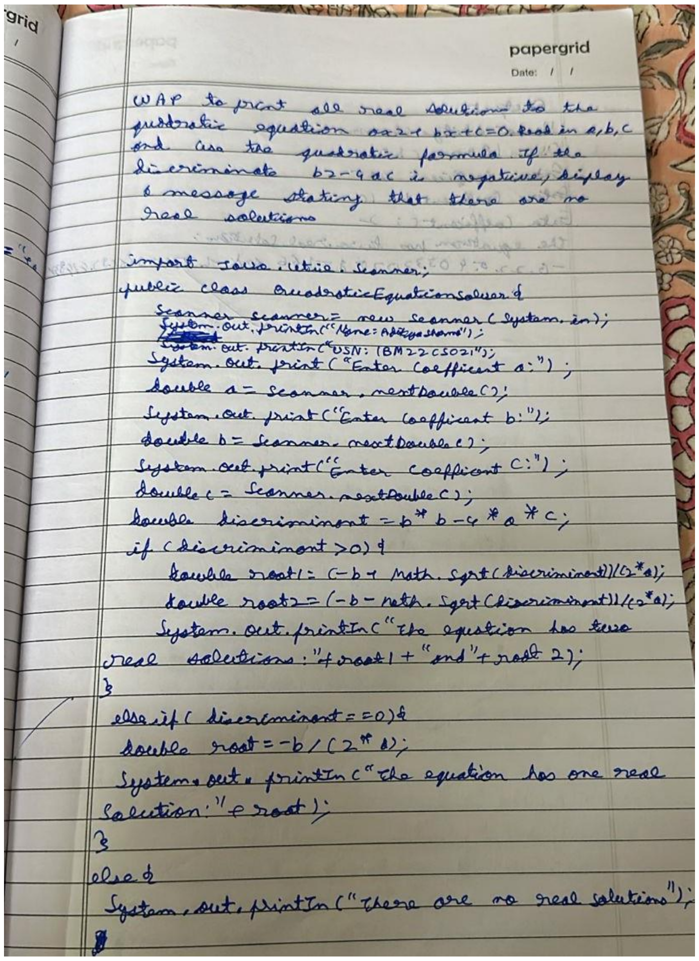
Index

		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.	
6	13/11/2024	Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.	24
7	20/11/2024	Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age=father’s age.	30
8	27/11/2024	Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.	34
9	27/11/2024	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.	36
10	27/11/2024	Demonstrate Inter process Communication and deadlock.	40

GitHub Link: https://github.com/aditya-sharma1230/Java-lab

<u>Program 1</u>
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.

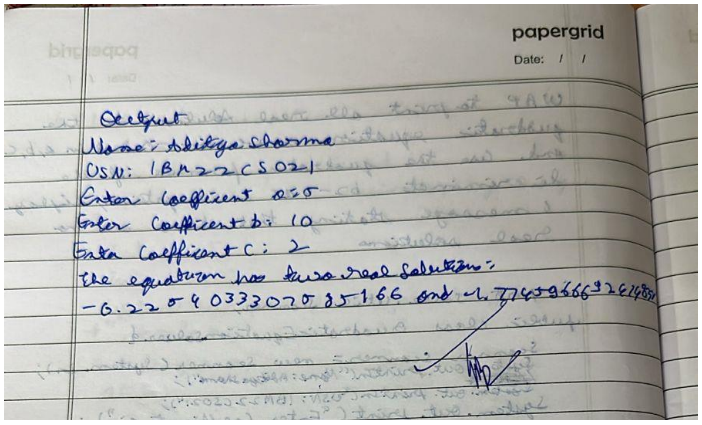
Algorithm:



papergrid
Date: / /

WAP to print 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

```
import Java.util.Scanner;
public class QuadraticEquationsolver {
   Scanner scanner = new Scanner(System.in);
   System.out.println("Name: Aditya sharma");
   System.out.println("USN: 1BM22CS021");
  System.out.print("Enter Coefficent a:");
  double a = Scanner.nextDouble();
  System.out.print("Enter coefficent b:");
  double b = Scanner.nextDouble();
  System.out.print("Enter coefficent C:");
  double c = Scanner.nextDouble();
 double discriminant = b*b-4*a*c;
 if (discriminant >0) {
     double root1 = (-b + Math.Sqrt(discriminant))/(2*a);
     double root2 = (-b - Math.Sqrt(discriminant))/(2*a);
    System.out.println("The equation has two
real solutions:"+root1 + "and"+root 2);
}
 else if( discriminant ==0){
 double root = -b/(2*a);
 System.out.println("The equation has one real
Solution:"+root);
}
else {
 System.out.println("There are no real solutions");
}
```



papergrid
Date: / /

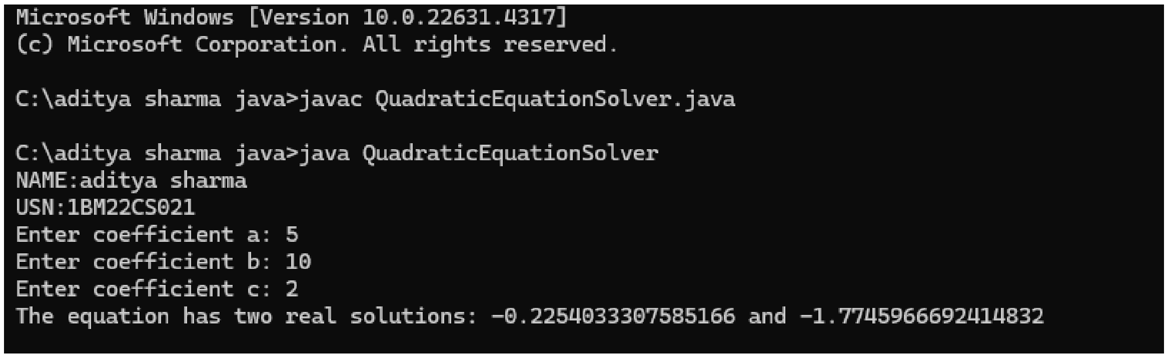
```
 Output
Name: Aditya sharma
USN: 1BM22CS021
Enter Coefficent a: 5
Enter Coefficent b: 10
Enter Coefficent C: 2
The equation has two real Solutions:
-0.2254033307585166 and -1.774596669241483
```

[illegible]

Code:

```
import java.util.Scanner;
public class QuadraticEquationSolver {
    public static void main(String[] args) {
            System.out.println("NAME:aditya sharma");
            System.out.println("USN:1BM22CS021");
        Scanner scanner = new Scanner(System.in);
        // Read coefficients a, b, and c
        System.out.print("Enter coefficient a: ");
        double a = scanner.nextDouble();
        System.out.print("Enter coefficient b: ");
        double b = scanner.nextDouble();
        System.out.print("Enter coefficient c: ");
        double c = scanner.nextDouble();
        // Calculate the discriminant
        double discriminant = b * b - 4 * a * c;
        // Determine the nature of the roots
        if (discriminant > 0) {
            // Two distinct real roots
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("The equation has two real solutions: " + root1 + " and " + root2);
        } else if (discriminant == 0) {
            // One real root
            double root = -b / (2 * a);
            System.out.println("The equation has one real solution: " + root);
        } else {
            // No real roots
            System.out.println("There are no real solutions.");
        }
        scanner.close();
    }
}
```

Output:

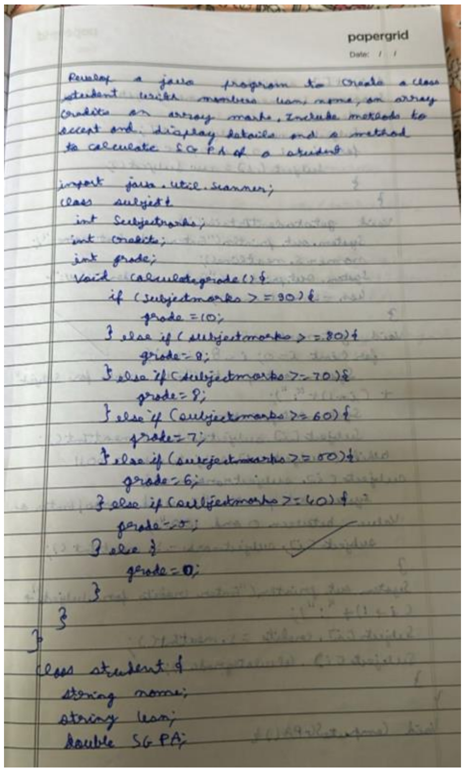


```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\aditya sharma java>javac QuadraticEquationSolver.java

C:\aditya sharma java>java QuadraticEquationSolver
NAME:aditya sharma
USN:1BM22CS021
Enter coefficient a: 5
Enter coefficient b: 10
Enter coefficient c: 2
The equation has two real solutions: -0.2254033307585166 and -1.7745966692414832
```

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:

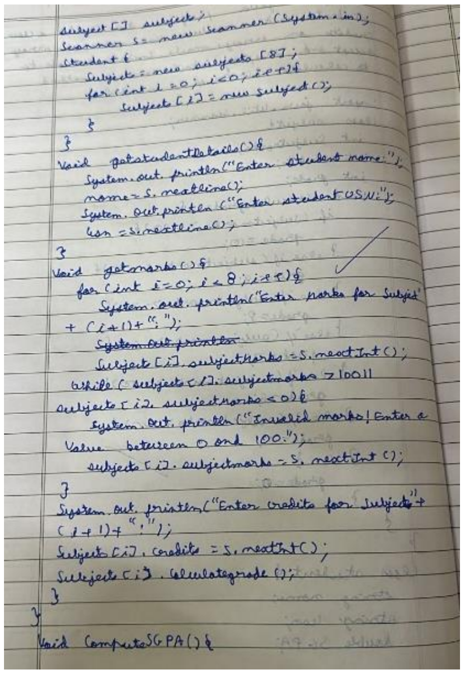


papergrid

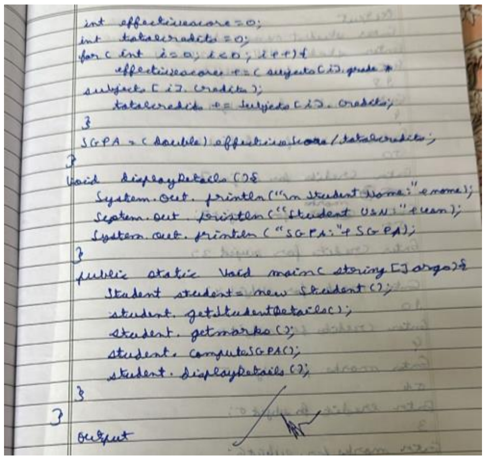
Date: / /

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

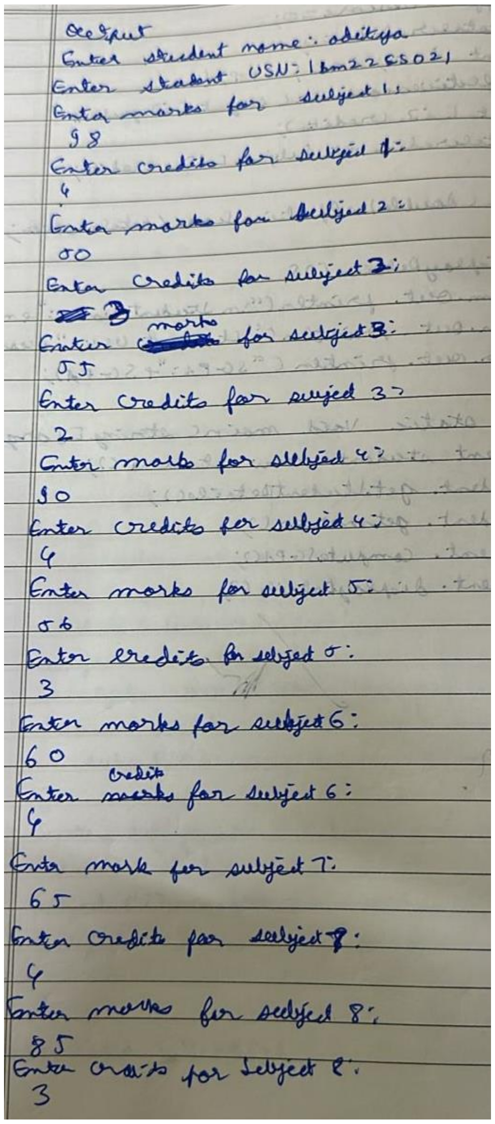
```
import java.util.scanner;
class subject{
  int Subjectmarks;
  int credits;
  int grade;
  Void calculategrade(){
      if (Subjectmarks >=90){
         grade =10;
      } else if (subjectmarks >=80){
         grade=9;
      } else if (subjectmarks >=70){
         grade=8;
      } else if (subjectmarks >=60){
         grade=7;
      } else if (subjectmarks >=50){
         grade=6;
      } else if (subjectmarks >=40){
        grade=5;
     } else {
         grade=0;
     }
  }
}
class student{
  string name;
  string usn;
  double SGPA;
```



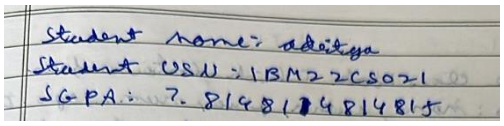
```
subject[] subjects;
Scanner s= new Scanner(System.in);
student{
   Subjects= new subjects [8];
   for(int i=0; i<8; i++){
       Subjects[i]= new subject();
   }
 }
Void getstudentDetails(){
   System.out.println("Enter student name:");
   name=s.nextline();
   System.out.println("Enter student USN:");
   usn =s.nextline();
}
Void getmarks(){
  for(int i=0; i<8; i++){
     System.out.println("Enter marks for Subject"
 + (i+1)+":");
     ~~System.out.println~~
     Subject[i].subjectmarks =S.nextInt();
  While( subjects[i].subjectmarks >100||
subjects[i].subjectmarks <0){
     System.out.println("Invalid marks! Enter a
 Value between 0 and 100:");
     subjects[i].subjectmarks =S.nextInt();
 }
System.out.println("Enter credits for Subject"+
(i+1)+":");
Subjects[i].credits =S.nextInt();
Subjects[i].calculategrade();
}
}
Void ComputeSGPA(){
```



```
int effectivescore=0;
int totalcredits =0;
for ( int i=0; i<n; i++){
      effectivescore +=( subjects[i].grade *
 subjects [ i]. credits);
     totalcredits += subjects [i]. credits;
 }
 SGPA = ( double) effectiveScore / totalcredits;
}
void displayDetails (){
   System.out. println("\n Student Name:"+name);
   System.out. println("Student USN:"+usn);
   System.out. println ("SGPA:"+ SGPA);
}
public static void main( string [] args){
    Student student= new Student ();
    student. getStudentDetails();
    student. getmarks ();
    student. computeSGPA();
    student. displayDetails ();
 }
}
Output
```



```
Output
Enter student name: aditya
Enter student USN: 1bm22CS021
Enter marks for subject 1:
 98
Enter credits for subject 1:
 4
Enter marks for subject 2:
 50
Enter credits for subject 2:
 3
Enter marks for subject 3:
 55
Enter credits for subject 3:
 2
Enter marks for subject 4:
 90
Enter credits for subject 4:
 4
Enter marks for subject 5:
 56
Enter credits for subject 5:
 3
Enter marks for subject 6:
 60
Enter credit for subject 6:
 4
Enter mark for subject 7:
 65
Enter Credits for subject 7:
 4
Enter marks for subject 8:
 85
Enter credits for subject 8:
 3
```



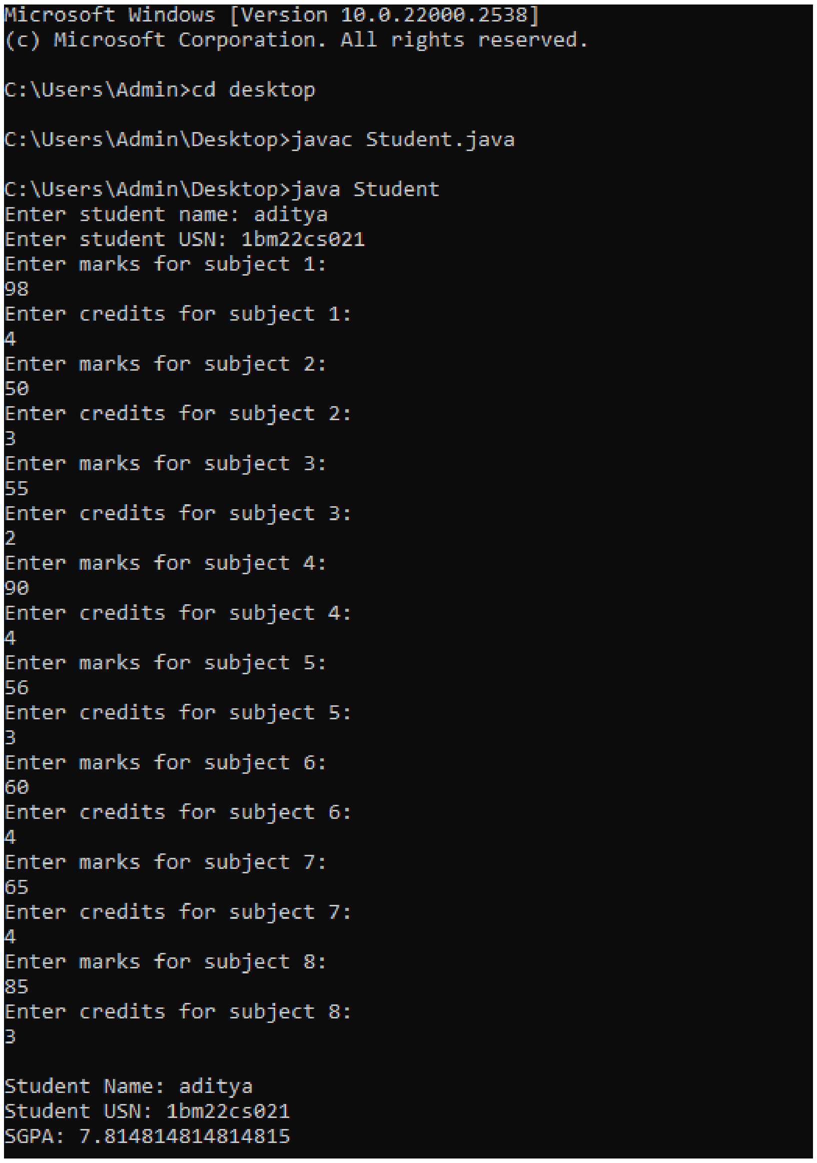
```
Student name: aditya
Student USN: 1BM22CS021
SGPA: 7.814814814814815
```

Code:

```
import java.util.Scanner;
// Class to represent a subject
class Subject {
    int subjectMarks;
    int credits;
    int grade;

    // Method to calculate grade based on marks
    void calculateGrade() {
        if (subjectMarks >= 90) {
            grade = 10;
        } else if (subjectMarks >= 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;  // Failed subject
        }
    }
}

// Class to represent a student
class Student {
    String name;
    String usn;
    double SGPA;
    Subject[] subjects;
    Scanner s = new Scanner(System.in);

    // Constructor
    Student() {
        subjects = new Subject[8];  // Array of 8 subjects
        for (int i = 0; i < 8; i++) {
            subjects[i] = new Subject();  // Initialize each subject
        }
    }

    // Method to get student details (name and usn)
    void getStudentDetails() {
```

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

// Method to get marks and credits for 8 subjects
void getMarks() {
    for (int i = 0; i < 8; i++) {
        System.out.println("Enter marks for subject " + (i + 1) + ": ");
        subjects[i].subjectMarks = s.nextInt();

        // Check if marks exceed 100
        while (subjects[i].subjectMarks > 100 || subjects[i].subjectMarks < 0) {
            System.out.println("Invalid marks! Enter a value between 0 and 100.");
            subjects[i].subjectMarks = s.nextInt();
        }

        System.out.println("Enter credits for subject " + (i + 1) + ": ");
        subjects[i].credits = s.nextInt();

        // Calculate grade for each subject based on marks
        subjects[i].calculateGrade();
    }
}

// Method to compute SGPA
void computeSGPA() {
    int effectiveScore = 0;
    int totalCredits = 0;

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

    SGPA = (double) effectiveScore / totalCredits;
}

// Method to display student details along with SGPA
void displayDetails() {
    System.out.println("\nStudent Name: " + name);
    System.out.println("Student USN: " + usn);
    System.out.println("SGPA: " + SGPA);
}

public static void main(String[] args) {
        // Create a Student object
        Student student = new Student();

        // Get student details, marks, and compute SGPA
        student.getStudentDetails();
        student.getMarks();
        student.computeSGPA();

        // Display the result
        student.displayDetails();
    }
}
```

Output:

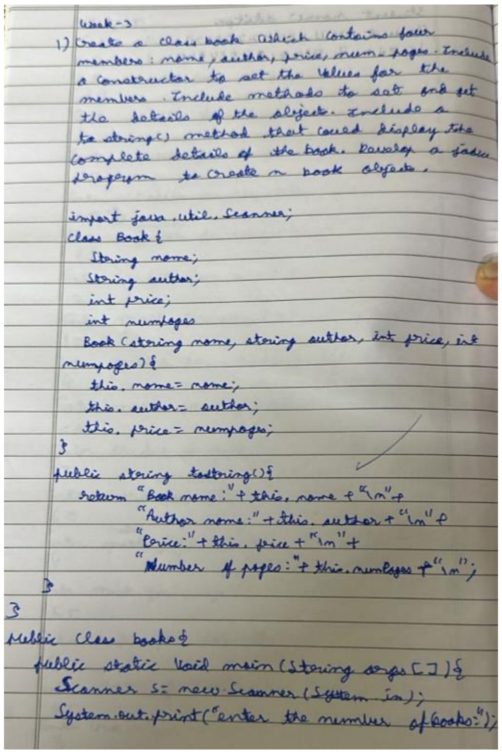


```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd desktop

C:\Users\Admin\Desktop>javac Student.java

C:\Users\Admin\Desktop>java Student
Enter student name: aditya
Enter student USN: 1bm22cs021
Enter marks for subject 1:
98
Enter credits for subject 1:
4
Enter marks for subject 2:
50
Enter credits for subject 2:
3
Enter marks for subject 3:
55
Enter credits for subject 3:
2
Enter marks for subject 4:
90
Enter credits for subject 4:
4
Enter marks for subject 5:
56
Enter credits for subject 5:
3
Enter marks for subject 6:
60
Enter credits for subject 6:
4
Enter marks for subject 7:
65
Enter credits for subject 7:
4
Enter marks for subject 8:
85
Enter credits for subject 8:
3

Student Name: aditya
Student USN: 1bm22cs021
SGPA: 7.814814814814815
```

<u>Program 3</u>

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.

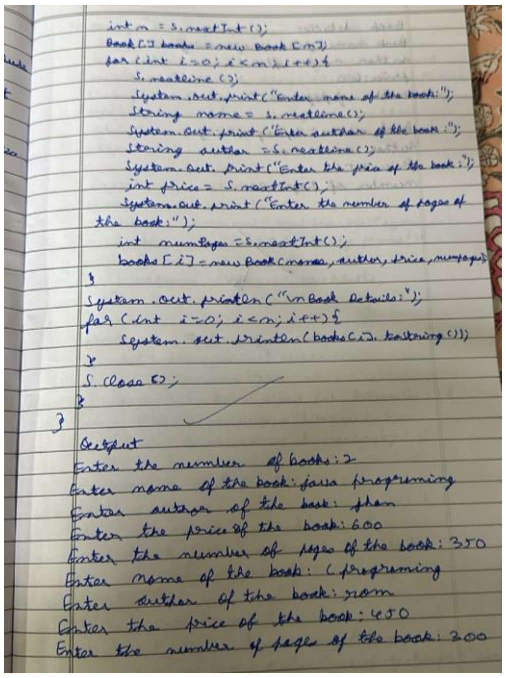
Algorithm:



Week-3

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

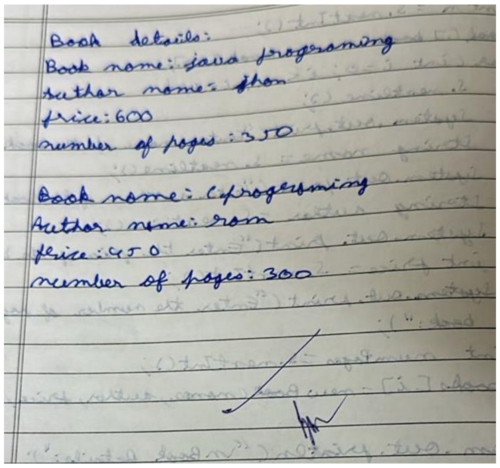
```
import java.util.Scanner;
class Book{
   String name;
   String author;
   int price;
   int numpages
   Book(string name, string author, int price, int
numpages){
   this. name= name;
   this. author= author;
   this. price= numpages;
 }
public string tostring(){
  return "Book name:"+ this. name +"\n"+
         "Author name:"+this. author+"\n"+
         "Price:"+ this. price + "\n"+
         "Number of pages:"+ this. numpages +"\n";
 }
}
public class books{
  public static void main(String args[]){
   Scanner s= new Scanner(System.in);
   System.out.print("enter the number of books:");
```



```
      int n = s.nextInt();
      Book[] books = new Book[n];
      for(int i=0; i<n; i++){
         s.nextline();
         System.out.print("Enter name of the book:");
         String name= s.nextline();
         System.out.print("Enter author of the book:");
         String author =s.nextline();
         System.out.print("Enter the price of the book:");
         int price= s.nextInt();
        System.out.print("Enter the number of pages of
   the book:");
        int numPages =s.nextInt();
        books[i]=new Book(name, author, price, numpages);
     }
     System.out.println("\nBook Details:");
     for(int i=0; i<n; i++){
        System.out.println(books[i].tostring());
     }
     S.close();
    }
 }
```

Output

```
Enter the number of books:2
Enter name of the book: java programing
Enter author of the book: jhon
Enter the price of the book:600
Enter the number of pages of the book:350
Enter name of the book: C programing
Enter author of the book: ram
Enter the price of the book:450
Enter the number of page of the book:300
```

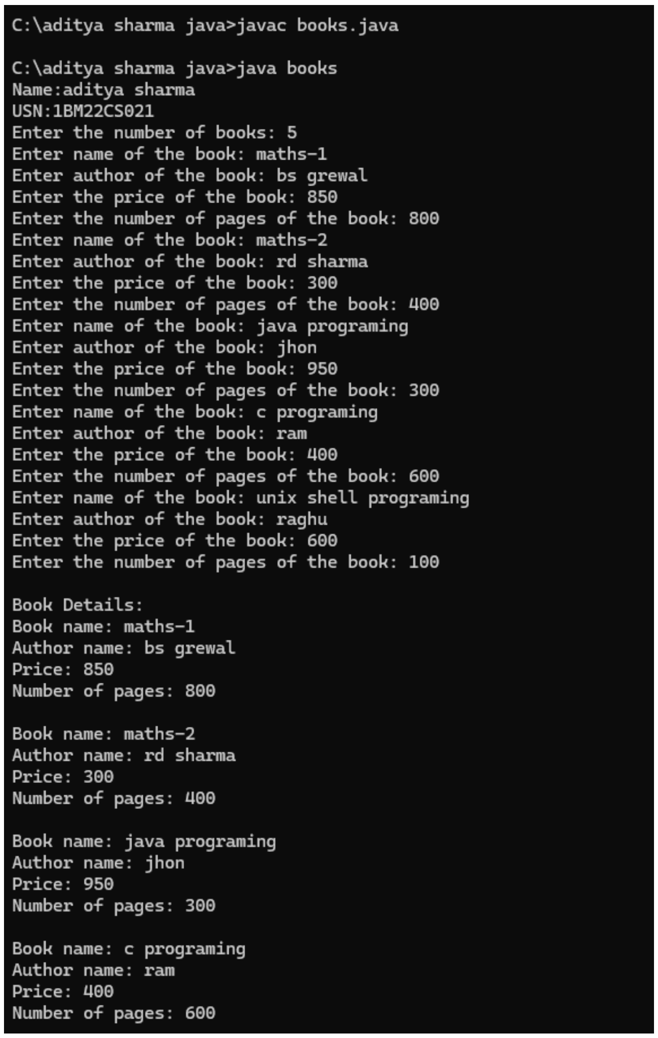


```
Book details:
Book name: java programing
Author name: jhon
price: 600
number of pages : 350

Book name: C programing
Author name: ram
price : 450
number of pages: 300
```

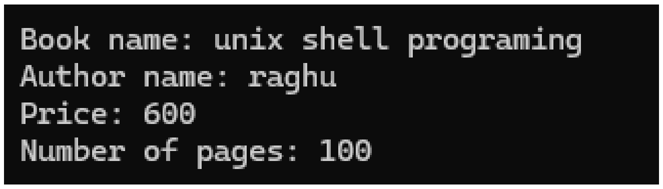
Code:

```
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";
    }
}

public class books {
    public static void main(String args[]) {
                System.out.println("Name: Aditya sharma");
        System.out.println("USN:1BM22CS021");
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        int n = s.nextInt();
        Book[] books = new Book[n];


        for (int i = 0; i < n; i++) {
            s.nextLine(); // Clear the buffer
            System.out.print("Enter name of the book: ");
            String name = s.nextLine();
            System.out.print("Enter author of the book: ");
            String author = s.nextLine();
            System.out.print("Enter the price of the book: ");
            int price = s.nextInt();
            System.out.print("Enter the number of pages of the book: ");
            int numPages = s.nextInt();


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


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

        s.close();
    }
}
```

Output:



```
C:\aditya sharma java>javac books.java

C:\aditya sharma java>java books
Name:aditya sharma
USN:1BM22CS021
Enter the number of books: 5
Enter name of the book: maths-1
Enter author of the book: bs grewal
Enter the price of the book: 850
Enter the number of pages of the book: 800
Enter name of the book: maths-2
Enter author of the book: rd sharma
Enter the price of the book: 300
Enter the number of pages of the book: 400
Enter name of the book: java programing
Enter author of the book: jhon
Enter the price of the book: 950
Enter the number of pages of the book: 300
Enter name of the book: c programing
Enter author of the book: ram
Enter the price of the book: 400
Enter the number of pages of the book: 600
Enter name of the book: unix shell programing
Enter author of the book: raghu
Enter the price of the book: 600
Enter the number of pages of the book: 100

Book Details:
Book name: maths-1
Author name: bs grewal
Price: 850
Number of pages: 800

Book name: maths-2
Author name: rd sharma
Price: 300
Number of pages: 400

Book name: java programing
Author name: jhon
Price: 950
Number of pages: 300

Book name: c programing
Author name: ram
Price: 400
Number of pages: 600
```

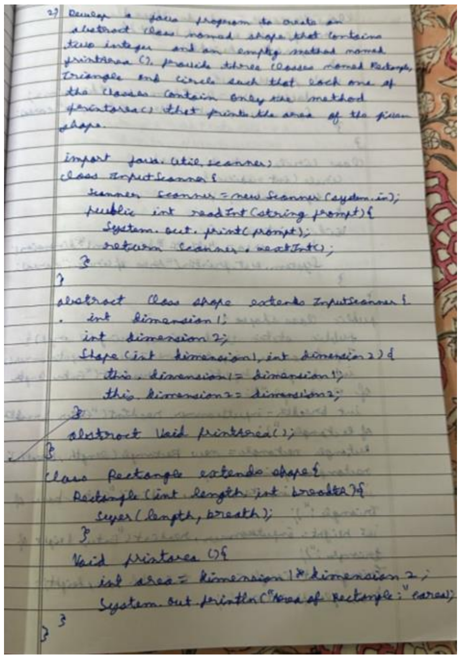


```
Book name: unix shell programing
Author name: raghu
Price: 600
Number of pages: 100
```

Program 4

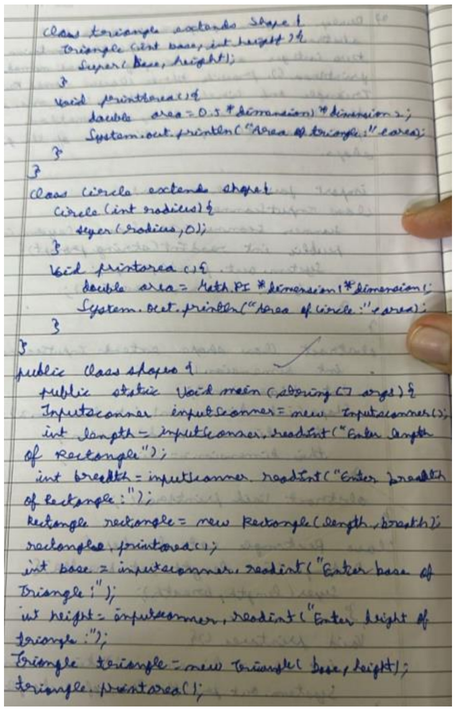
Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea(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

Algorithm:

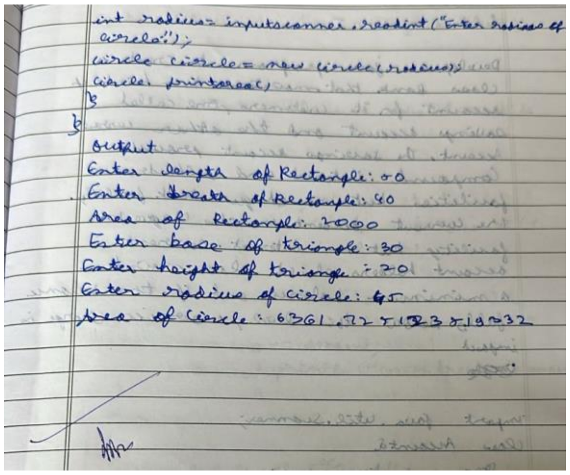


27) Develop a java program to create an abstract class named shape that contains two integer and an empty method named printArea (). Provide three classes named Rectangle, Triangle and circle such that each one of the classes contain only the method printarea() that prints the area of the given shape.

```
import java.util.scanner;
class InputScanner{
    scanner scanner = new Scanner(system.in);
    public int readInt(string prompt){
        System.out.print(prompt);
        return scanner.nextInt();
    }
}
abstract class shape extends InputScanner{
    int dimension1;
    int dimension2;
    Shape(int dimension1, int dimension2){
        this.dimension1= dimension1;
        this.dimension2= dimension2;
    }
    abstract void printArea();
}
class Rectangle extends shape{
    Rectangle(int length, int breadth){
        super(length, breath);
    }
    void printarea (){
        int area= dimension1* dimension2;
        System.out.println("Area of Rectangle:"+area);
    }
}
```



```
class triangle extends Shape{
    Triangle(int base, int height){
        Super(Base, height);
    }
    void printArea(){
        double area=0.5*dimension1*dimension2;
        System.out.println("Area of triangle:"+area);
    }
}
class circle extends shape{
    circle(int radius){
        super(radius,0);
    }
    void printarea(){
        double area= Math.PI *dimension1*dimension1;
        System.out.println("Area of circle:"+area);
    }
}
public class shapes{
    public static void main(string[] args){
    Inputscanner inputscanner= new Inputscanner();
    int length= inputscanner.readInt("Enter length
of Rectangle");
    int breadth= inputscanner.readInt("Enter breadth
of Rectangle:");
Rectangle rectangle= new Rectangle(length, breath);
rectangle.printarea();
int base = inputscanner.readint("Enter base of
Triangle:");
int height= inputscanner.readint("Enter height of
triangle:");
Triangle triangle= new Triangle(base, height);
triangle.printarea();
```

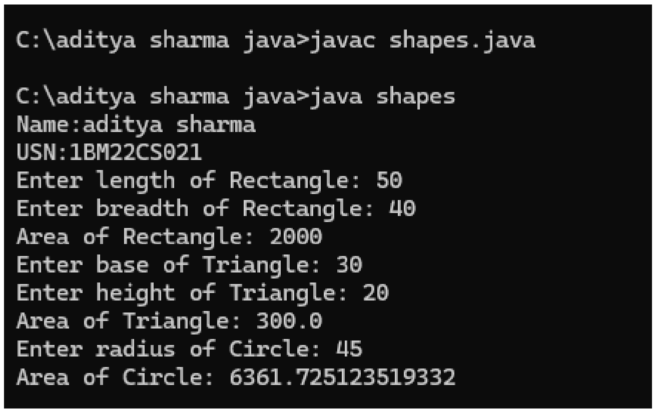


```
int radius= inputscanner.readint("Enter radius of circle:");
circle circle= new circle(radius);
circle.printarea()
}
}
Output
Enter length of Rectangle: 50
Enter breath of Rectangle: 40
Area of Rectangle: 2000
Enter base of triangle: 30
Enter height of triangle: 20
Enter radius of circle: 45
Area of Circle: 6361.725123519232
```

Code:

```
import java.util.Scanner;

// Class to handle user input
class InputScanner {
    Scanner = new Scanner(System.in);

    // Method to read integer input
    public int readInt(String prompt) {
        System.out.print(prompt);
        return scanner.nextInt();
    }
}

// Abstract class Shape
abstract class Shape extends InputScanner {
    int dimension1; // Can represent length or radius
    int dimension2; // Can represent breadth or height (not used in Circle)

    // Constructor to initialize dimensions
    Shape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }

    // Abstract method to print area
    abstract void printArea();
}

// Class Rectangle
class Rectangle extends Shape {
    Rectangle(int length, int breadth) {
        super(length, breadth);
    }

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

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

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

// Class Circle
class Circle extends Shape {
    Circle(int radius) {
        super(radius, 0); // dimension2 is not used
    }

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

// Main class
public class shapes {
    public static void main(String[] args) {
                System.out.println("Name: Aditya sharma");
        System.out.println("USN:1BM22CS021");

        InputScanner inputScanner = new InputScanner();

        // Rectangle
        int length = inputScanner.readInt("Enter length of Rectangle: ");
        int breadth = inputScanner.readInt("Enter breadth of Rectangle: ");
        Rectangle rectangle = new Rectangle(length, breadth);
        rectangle.printArea();

        // Triangle
        int base = inputScanner.readInt("Enter base of Triangle: ");
        int height = inputScanner.readInt("Enter height of Triangle: ");
        Triangle triangle = new Triangle(base, height);
        triangle.printArea();

        // Circle
        int radius = inputScanner.readInt("Enter radius of Circle: ");
        Circle circle = new Circle(radius);
        circle.printArea();
    }
}
```

Output:



```
C:\aditya sharma java>javac shapes.java

C:\aditya sharma java>java shapes
Name:aditya sharma
USN:1BM22CS021
Enter length of Rectangle: 50
Enter breadth of Rectangle: 40
Area of Rectangle: 2000
Enter base of Triangle: 30
Enter height of Triangle: 20
Area of Triangle: 300.0
Enter radius of Circle: 45
Area of Circle: 6361.725123519332
```

<u>Program 5</u>

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.

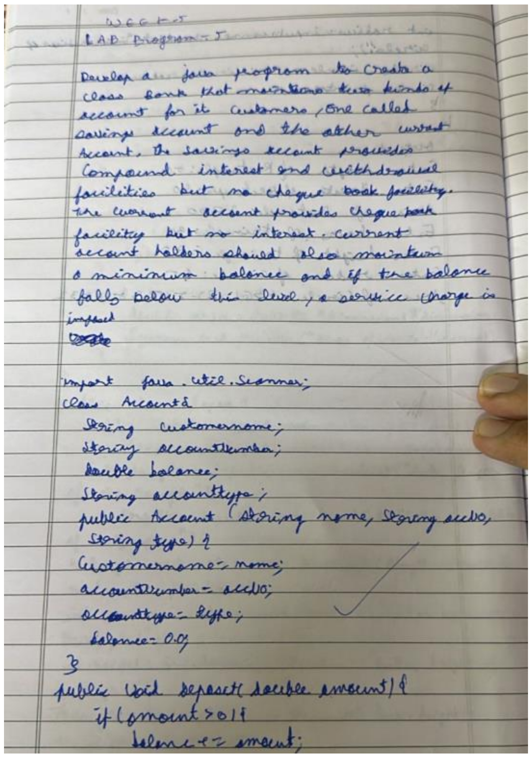
b) Display the balance.

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:

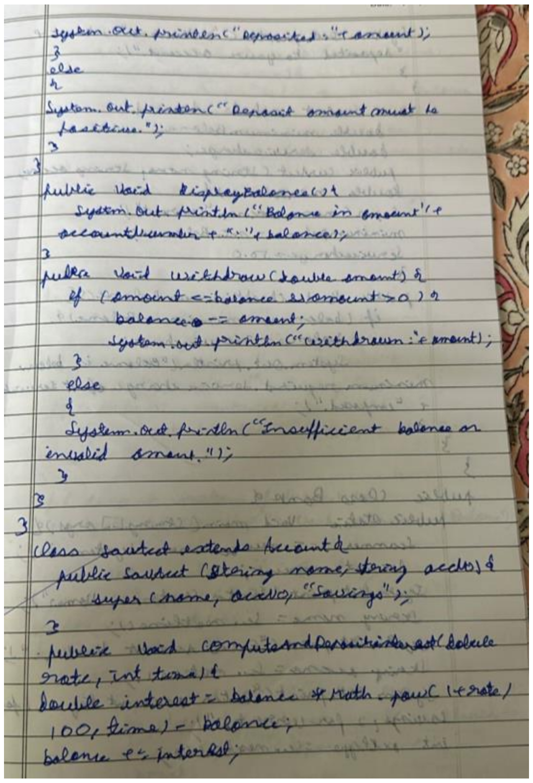


WEEK-5
LAB Program-5

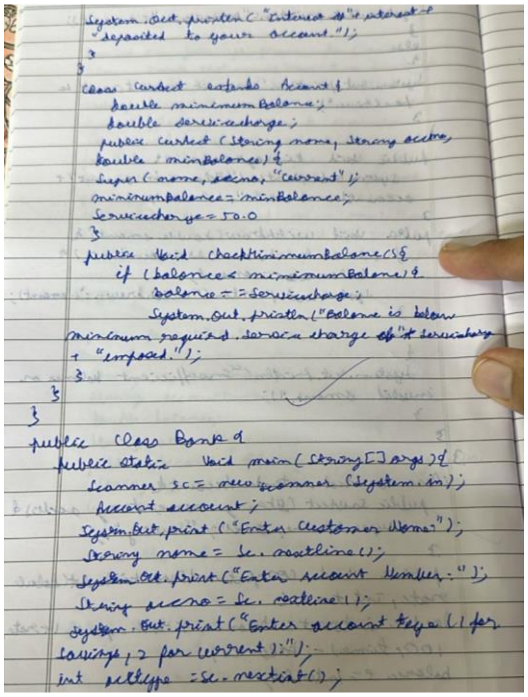
Develop a java program to create a class Bank that maintains two kinds of account for its customers, One called savings account and the other current Account. the savings account provides Compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book 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

~~[illegible]~~

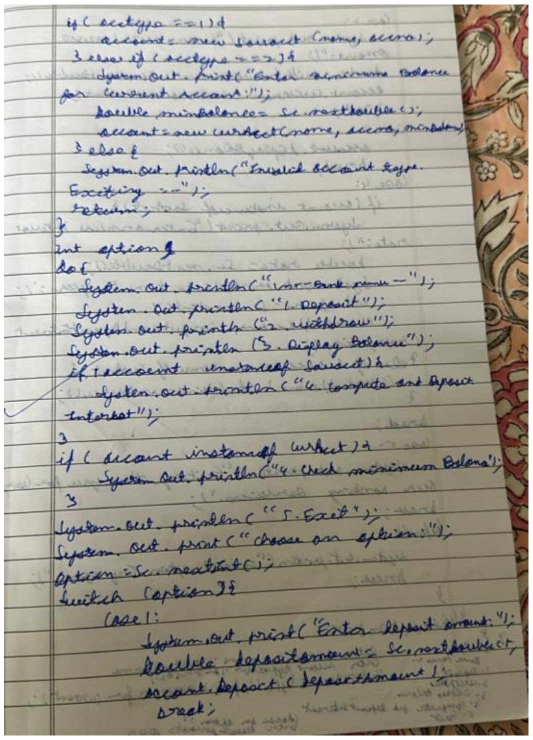
```
import java.util.Scanner;
class Account{
  String customername;
  String accountNumber;
  double balance;
  String accounttype;
  public Account (String name, String accNo,
   String type) {
  Customername= name;
  accountNumber= accNo;
  accounttype= type;
   balance= 0.0;
 }
public void deposit( double amount) {
    if (amount >0) {
        balance += amount;
```



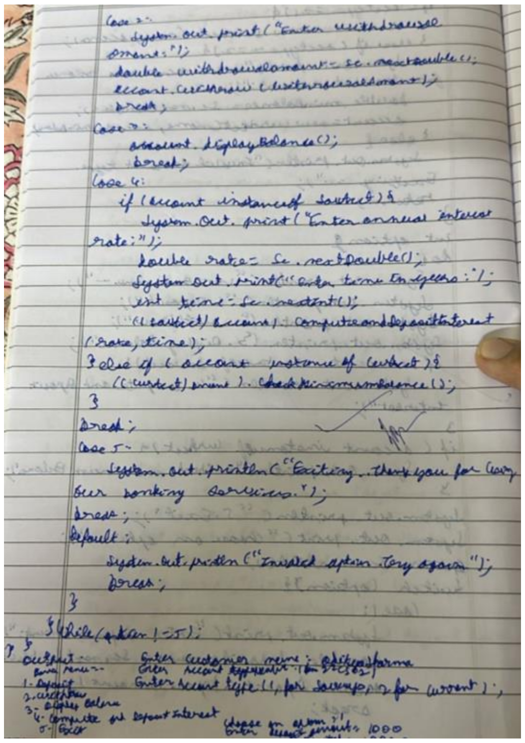
```
system.out.println("Deposited :"+amount);
 }
else
{
System.out.println("Deposit amount must be
 positive.");
 }
}
public void displayBalance(){
   System.out.println("Balance in amount"+
 accountNumber+":"+balance);
}
public void withdraw(double amount) {
  if (amount <=balance &&amount>0 ){
      balance -= amount;
      system.out.println("withdrawn:"+amount);
  }
  else
  {
  System.out.println("Insufficient balance or
invalid amount.");
  }
}
}
class savAcct extends Account{
  public savAcct (String name, String accNo){
     super(name, accNo, "Savings");
  }
 public void computeAndDepositInterest(double
rate, int time){
double interest= balance * Math.pow(1+rate/
 100, time)- balance;
balance += interest;
```



```
   System.out.println("Interest "+interest+
   "deposited to your account.");
  }
 }
 class curAcct extends Account{
    double minimumBalance;
    double servicecharge;
   public curAcct (String name, String accno,
  double minBalance){
  super(name, accno, "current");
  minimumBalance= minBalance;
  servicecharge=50.0
  }
  public void checkMinimumBalance(){
     if (balance< minimumBalance){
        balance-=servicecharge;
           System.out.println("Balance is below
 minimum required. service charge ~~of~~"+ servicecharge
 + "imposed.");
  }
 }
}
public class Bank{
  public static void main(String[] args){
     Scanner sc= new Scanner (System.in);
      Account account;
    System.out.print ("Enter Customer Name:");
    String name= sc.nextLine();
    System.out.print("Enter account Number:");
    String accno= sc.nextLine();
   System.out.print("Enter account type (1 for
 savings, 2 for current):");
 int accType =sc.nextInt();
```



```
if ( acctype ==1){
    account= new SavAcct (name, accno);
 } else if ( acctype ==2){
    System.out. print("Enter minimum Balance
for current account:");
    double minBalance= sc.nextDouble();
    account=new CurAcct(name, accno, minBalance);
 } else {
  System.out. println("Invalid account type.
 Exiting ...");
  return;
}
int option;
do{
  System.out. println("--- Bank menu --");
  System. out. println ("1. Deposit");
  System.out. println ("2. withdraw");
  System.out. println ("3. Display Balance");
  if (account instanceof SavAcct){
     System.out. println ("4. Compute and Deposit
 Interest");
}
if ( account instanceof CurAcct ){
     System.Out. println("4. Check minimum Balance");
 }
System.out. println ("5. Exit");
System. out. print ("choose an option:");
option =sc. nextInt();
switch (option){
     case 1:
        System.out. print("Enter deposit amount:");
        double depositAmount= sc.nextDouble();
        account. Deposit ( depositAmount );
        break;
```



```
     case 2:
        System. out. print ("Enter withdrawal
        amount:");
        double withdrawalamount = sc. nextDouble();
        account. withdraw (withdrawalamount);
        break;
     case 3:
        account. displayBalance();
        break;
     case 4:
       if (account instanceof SavAcct){
          System.Out. print ("Enter annual interest
     rate:");
          double rate= sc. nextDouble();
          System.out. print("Enter time in years :");
          int time= sc. nextInt();
          ((SavAcct) account). computeandDepositInterest
     (rate, time);
     } else if ( account instance of CurAcct ){
        ((CurAcct) account ). CheckMinimumBalance();
     }
     break;
     case 5:
        System. out. println ("Exiting.. thank you for using
     our banking services.");
     break;
     Default:
        System. out. println ("Invalid option. Try again");
        break;
     }
   }while(option != 5);
 }
output:        Enter customer name : Aditya sharma
 Bank menu:-   Enter account appenable : [illegible]
1- Deposit     Enter account type (1, for savings, 2 for current ) :
2. withdraw
3- display Balance
4. Compute and deposit interest
                    choose an option :1
 5- Exit            Enter deposit amount: 1000
```

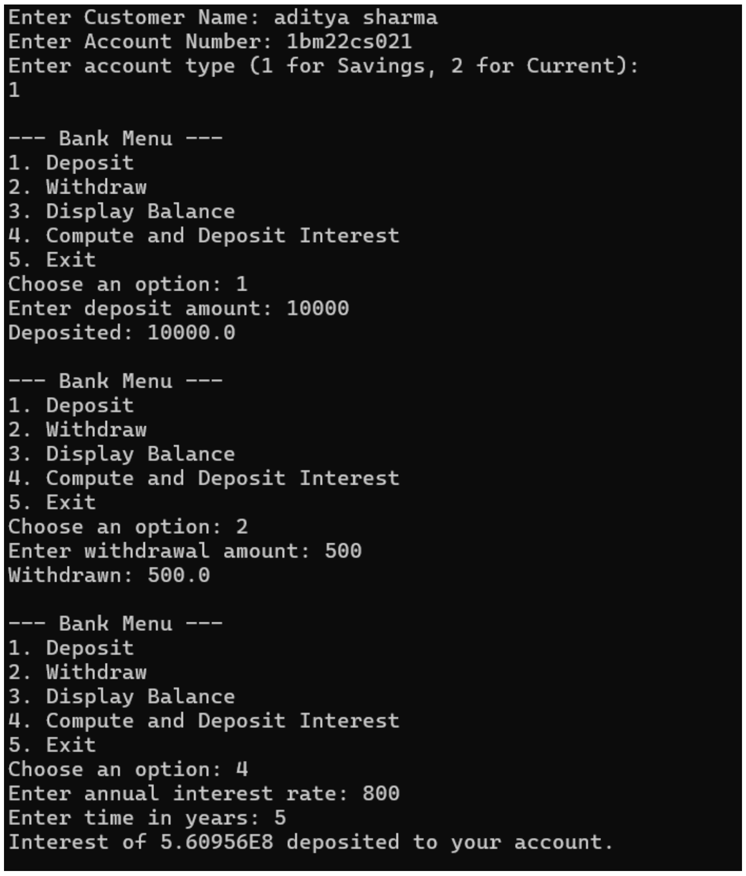
Code:

```
import java.util.Scanner;
// Base class Account
class Account {
    String customerName;
    String accountNumber;
    double balance;
    String accountType;

    // Constructor to initialize the Account details
    public Account(String name, String accNo, String type) {
        customerName = name;
        accountNumber = accNo;
        accountType = type;
        balance = 0.0;
    }

    // Method to accept deposit and update balance
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }

    // Method to display the balance
    public void displayBalance() {
        System.out.println("Balance in account " + accountNumber + ": " + balance);
    }

    // Method to permit withdrawal and update balance
    public void withdraw(double amount) {
        if (amount <= balance && amount > 0) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance or invalid amount.");
        }
    }
}

// Class for Savings Account derived from Account
class SavAcct extends Account {
    // Constructor for Savings Account
    public SavAcct(String name, String accNo) {
        super(name, accNo, "Savings");
    }

    // Method to compute and deposit interest (compound interest)
    public void computeAndDepositInterest(double rate, int time) {
        double interest = balance * Math.pow(1 + rate / 100, time) - balance;
        balance += interest;
        System.out.println("Interest of " + interest + " deposited to your account.");
    }
}

// Class for Current Account derived from Account
class CurAcct extends Account {
    double minimumBalance;
    double serviceCharge;

    // Constructor for Current Account
    public CurAcct(String name, String accNo, double minBalance) {
        super(name, accNo, "Current");
        minimumBalance = minBalance;
        serviceCharge = 50.0; // Assume service charge is 50 units if balance falls below minimum
    }

    // Method to check minimum balance, impose penalty if necessary
    public void checkMinimumBalance() {
        if (balance < minimumBalance) {
            balance -= serviceCharge;
            System.out.println("Balance is below minimum required. Service charge of " + serviceCharge
+ " imposed.");
        }
    }
}

// Main class to test the Bank System
public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Account account;

        // Create a new account (example)
        System.out.print("Enter Customer Name: ");
        String name = sc.nextLine();

        System.out.print("Enter Account Number: ");
        String accNo = sc.nextLine();
```

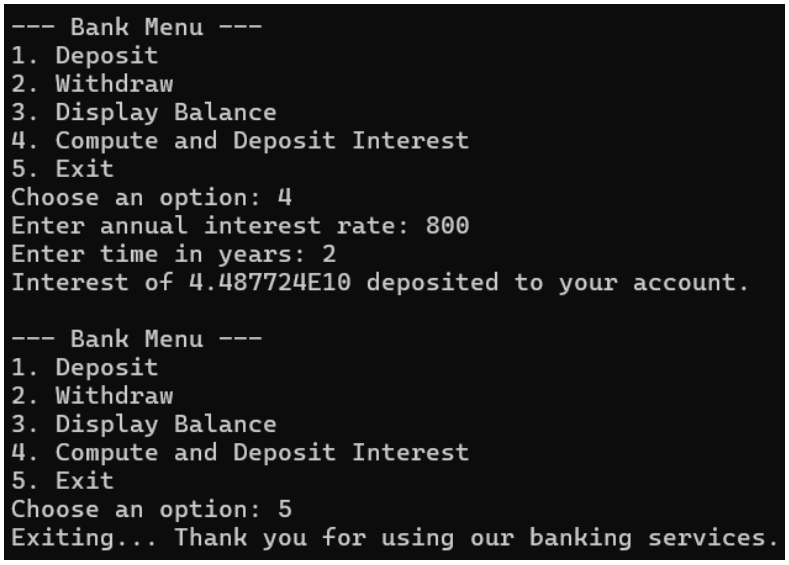
```
System.out.println("Enter account type (1 for Savings, 2 for Current): ");
int accType = sc.nextInt();

// Create account based on type
if (accType == 1) {
    account = new SavAcct(name, accNo);
} else if (accType == 2) {
    System.out.print("Enter Minimum Balance for Current Account: ");
    double minBalance = sc.nextDouble();
    account = new CurAcct(name, accNo, minBalance);
} else {
    System.out.println("Invalid account type. Exiting...");
    return;
}

// Perform operations on the account
int option;
do {
    System.out.println("\n--- Bank Menu ---");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display Balance");
    if (account instanceof SavAcct) {
        System.out.println("4. Compute and Deposit Interest");
    }
    if (account instanceof CurAcct) {
        System.out.println("4. Check Minimum Balance");
    }
    System.out.println("5. Exit");
    System.out.print("Choose an option: ");
    option = sc.nextInt();

    switch (option) {
        case 1:
            System.out.print("Enter deposit amount: ");
            double depositAmount = sc.nextDouble();
            account.deposit(depositAmount);
            break;
        case 2:
            System.out.print("Enter withdrawal amount: ");
            double withdrawalAmount = sc.nextDouble();
            account.withdraw(withdrawalAmount);
            break;
        case 3:
            account.displayBalance();
            break;
```

```
                case 4:
                    if (account instanceof SavAcct) {
                        System.out.print("Enter annual interest rate: ");
                        double rate = sc.nextDouble();
                        System.out.print("Enter time in years: ");
                        int time = sc.nextInt();
                        ((SavAcct) account).computeAndDepositInterest(rate, time);
                    } else if (account instanceof CurAcct) {
                        ((CurAcct) account).checkMinimumBalance();
                    }
                    break;
                case 5:
                    System.out.println("Exiting... Thank you for using our banking services.");
                    break;
                default:
                    System.out.println("Invalid option. Try again.");
                    break;
            }
        } while (option != 5);
    }
}
```

Output:



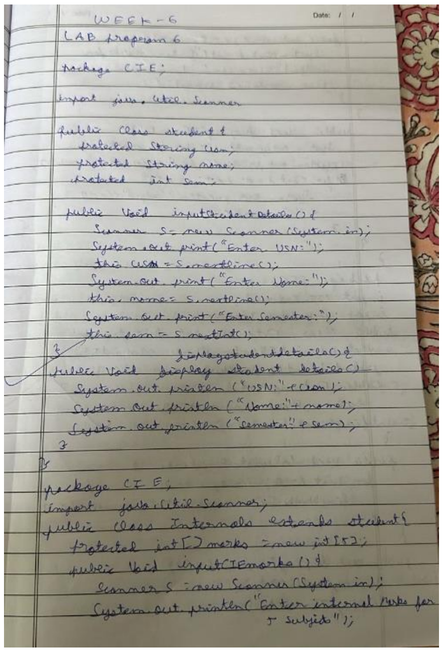
```
Enter Customer Name: aditya sharma
Enter Account Number: 1bm22cs021
Enter account type (1 for Savings, 2 for Current):
1

--- Bank Menu ---
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
Choose an option: 1
Enter deposit amount: 10000
Deposited: 10000.0

--- Bank Menu ---
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
Choose an option: 2
Enter withdrawal amount: 500
Withdrawn: 500.0

--- Bank Menu ---
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
Choose an option: 4
Enter annual interest rate: 800
Enter time in years: 5
Interest of 5.60956E8 deposited to your account.
```


```
--- Bank Menu ---
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
Choose an option: 4
Enter annual interest rate: 800
Enter time in years: 2
Interest of 4.487724E10 deposited to your account.

--- Bank Menu ---
1. Deposit
2. Withdraw
3. Display Balance
4. Compute and Deposit Interest
5. Exit
Choose an option: 5
Exiting... Thank you for using our banking services.
```

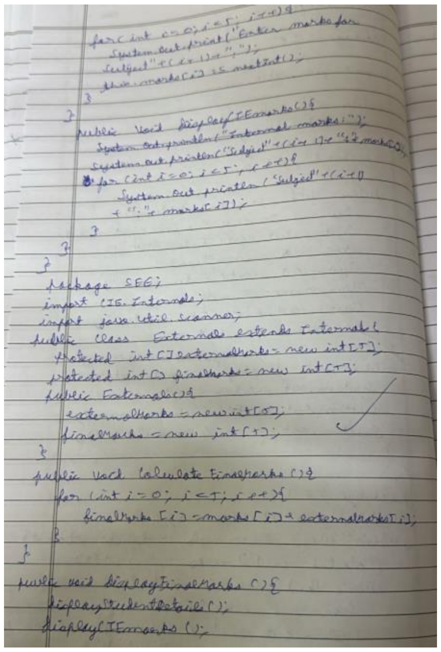
Program 6

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

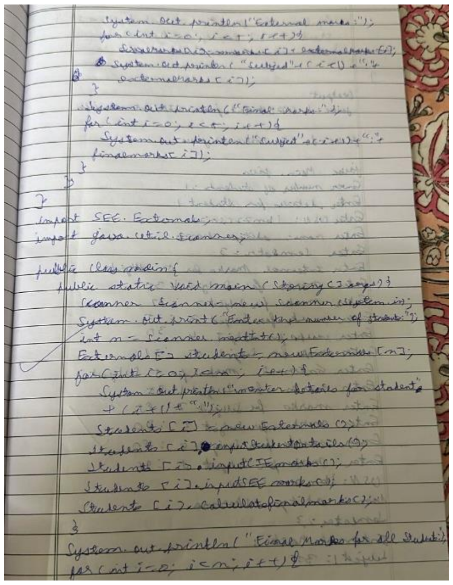
Algorithm:



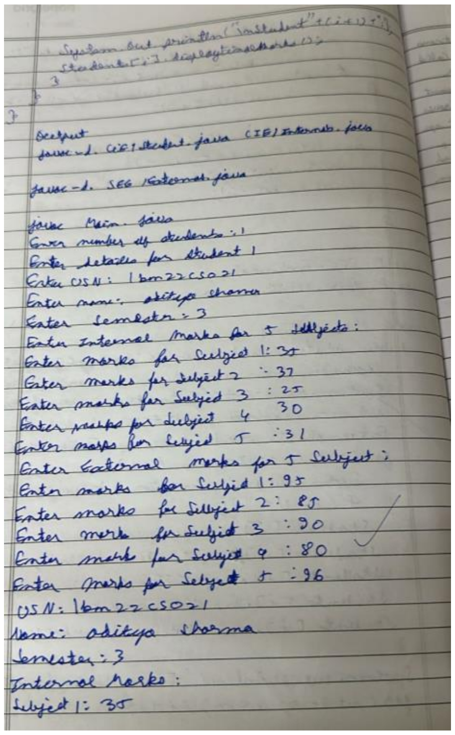
```
WEEK-6                                    Date: / /
LAB program 6

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:");
      this.usn = S.nextLine();
      System.out.print("Enter Name:");
      this.name = S.nextLine();
      System.out.print("Enter Semester:");
      this.sem = S.nextInt();
   }
                     displaystudentdetails() {
 public void display student details()
    System.out.println ("USN:"+usn);
    System.out.println ("Name:"+name);
    System.out.println ("Semester:"+sem);
  }
}
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 internal marks for
                         5 subjects");
```



```
      for(int i=0; i<5; i++) {
        System.out.print("Enter marks for
       Subject"+(i+1)+":");
        this.marks[i] = S.nextInt();
      }
   }
  public void displayCIEmarks() {
    System.out.println("Internal marks:");
    System.out.println("Subject"+(i+1)+":"+marks[i]);
    System.out.println("Subject"+(i+1)){
    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[] externalMarks = new int[5];
  protected int[] finalMarks = new int[5];
  public Externals(){
    externalMarks = new int[5];
    finalMarks = new int[5];
  }
 public void calculateFinalMarks(){
   for (int i=0; i<5; i++){
      finalMarks[i] = marks[i] + externalMarks[i];
   }
 }
public void displayFinalMarks(){
  displayStudentDetails();
  displayCIEmarks();
```



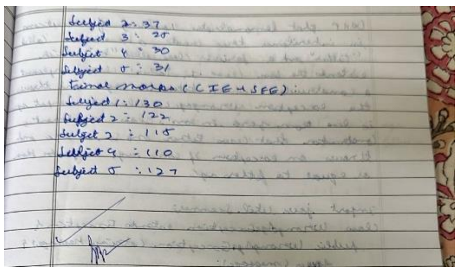
```
        System.out.println("External marks:");
        for (int i=0; i<5; i++){
            finalmarks[i] = marks[i] + externalmarks[i];
            System.out.println( "Subject" + (i+1) + ":" +
            externalmarks[i]);
        }
      System.out.println("Final marks:");
      for (int i=0; i<5; i++){
        System.out.println("Subject"+(i+1)+":"+
        finalmarks[i]);
      }
    }
}
import SEE.Externals;
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();
    Externals[] students = new Externals[n];
    for(int i=0; i<n; i++){
      System.out.println("\nEnter details for student"
      + (i+1) + ":");
      Students[i] = new Externals();
      students[i].inputStudentDetails();
      Students[i].inputCIEmarks();
      Students[i].inputSEEmarks();
      Students[i].calculatefinalmarks();
    }
    System.out.println("Final Marks for all Students:");
    for (int i=0; i<n; i++){
```



```
      System.out.println("\nStudent"+(i+1)+":");
      students[i].displayfinalmarks();
    }
  }
}
```

Output

```
javac -d . CIE/Student.java CIE/Internals.java

javac -d . SEE/Externals.java

javac Main.java
Enter number of students:1
Enter details for Student 1
Enter USN: 1bm22cs021
Enter name: aditya sharma
Enter Semester: 3
Enter Internal Marks for 5 subjects:
Enter marks for Subject 1: 35
Enter marks for Subject 2 : 37
Enter marks for Subject 3 : 25
Enter marks for Subject 4  30
Enter marks for Subject 5 : 31
Enter External marks for 5 Subject:
Enter marks for Subject 1: 95
Enter marks for Subject 2: 85
Enter marks for Subject 3 : 90
Enter mark for Subject 4 : 80
Enter marks for Subject 5 : 96
USN: 1bm22cs021
Name: aditya sharma
Semester: 3
Internal Marks:
Subject 1: 35
```



```
Subject 2: 37
Subject 3 : 25
Subject 4 : 30
Subject 5 : 31
Final marks (CIE+SEE):
Subject 1: 130
Subject 2 : 122
Subject 3  : 115
Subject 4  : 110
Subject 5  : 127
```

Code:

CIE Package – Student.java

```
package CIE;

import java.util.Scanner;

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

    // Method to input student details
    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();
    }

    // Method to display student details
    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
```

CIE Package – Internals.java

```
package CIE;

import java.util.Scanner;

public class Internals extends Student {
    protected int[] marks = new int[5];  // Array to store internal marks

    // Method to input CIE marks for 5 subjects
    public void inputCIEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Internal marks for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = s.nextInt();
        }
    }

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

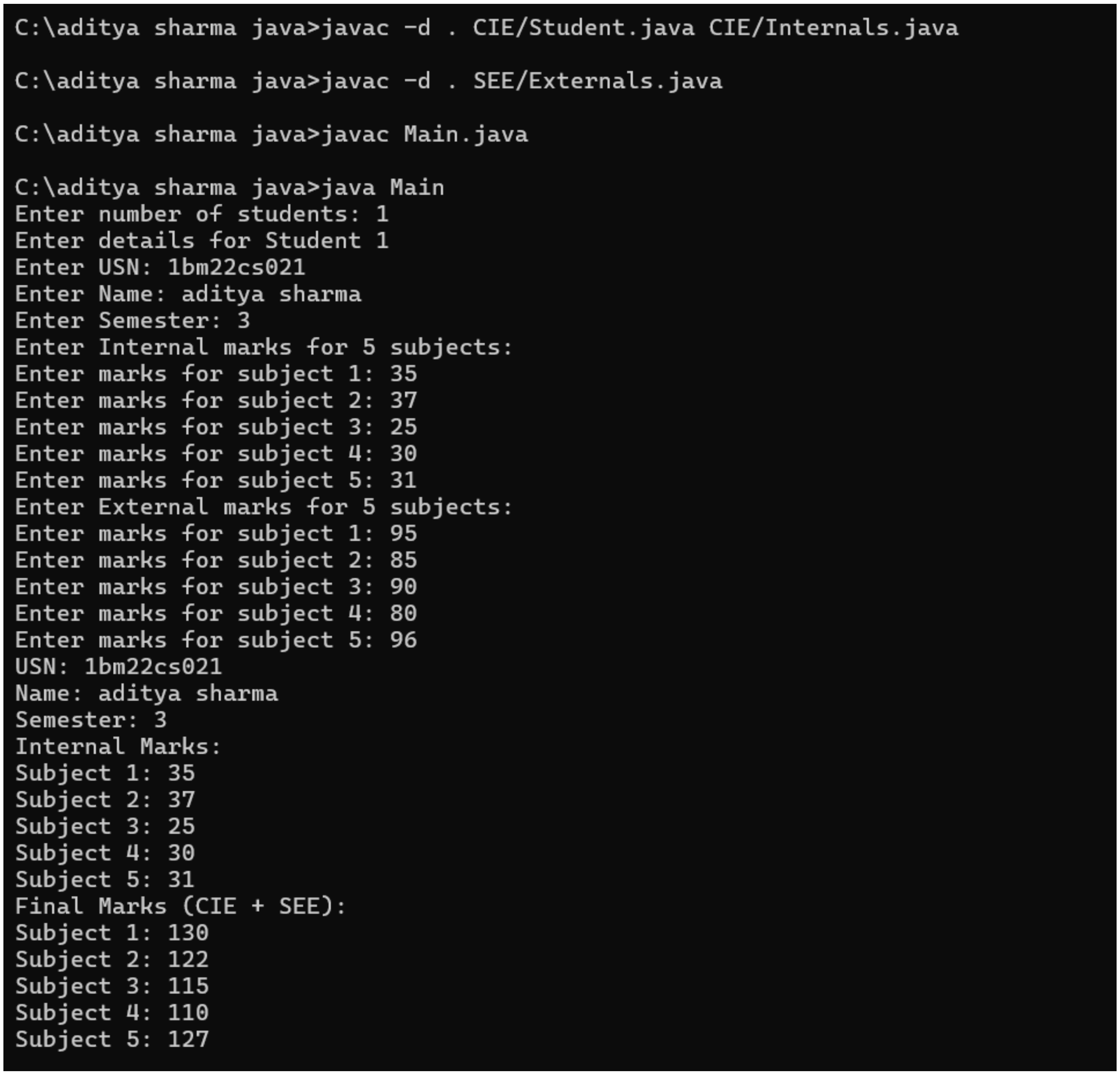
SEE Package – Externals.java

```
package SEE;

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

public class Externals extends Internals {
    protected int[] externalMarks = new int[5];  // Array to store external marks
    protected int[] finalMarks = new int[5];     // Array to store final marks

    // Constructor to initialize arrays
    public Externals() {
        externalMarks = new int[5];
        finalMarks = new int[5];
    }

    // Method to input SEE marks for 5 subjects
    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 marks for subject " + (i + 1) + ": ");
            externalMarks[i] = s.nextInt();
        }
    }

    // Method to calculate final marks (CIE + SEE)
    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = marks[i] + externalMarks[i];  // Add internal and external marks
        }
    }
    // Method to display final marks
    public void displayFinalMarks() {
        displayStudentDetails();
        displayCIEmarks();
        System.out.println("Final Marks (CIE + SEE):");
        for (int i = 0; i < 5; i++) {
            System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
        }
    }
}
```

Main.java

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

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

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

        // Array of Externals objects to handle multiple students
        Externals[] students = new Externals[n];

        for (int i = 0; i < n; i++) {
            students[i] = new Externals();
            System.out.println("Enter details for Student " + (i + 1));
            students[i].inputStudentDetails();  // Input student details
            students[i].inputCIEmarks();        // Input CIE marks
            students[i].inputSEEmarks();        // Input SEE marks
            students[i].calculateFinalMarks();  // Calculate final marks
            students[i].displayFinalMarks();    // Display final marks
        }
    }
}
```

Output:

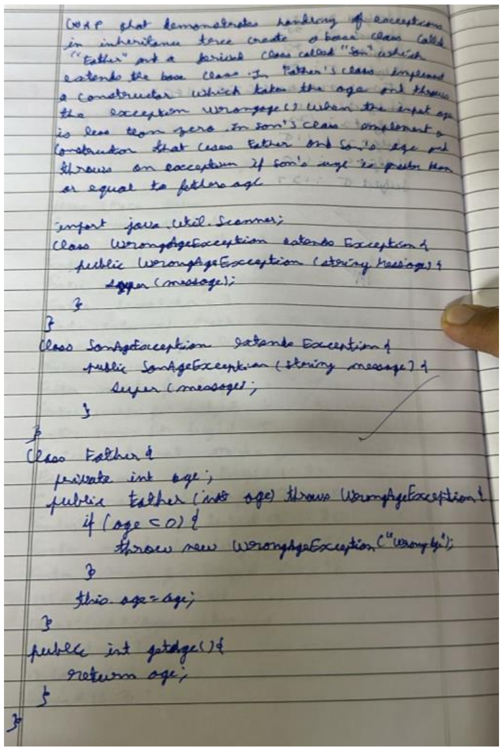


```
C:\aditya sharma java>javac -d . CIE/Student.java CIE/Internals.java

C:\aditya sharma java>javac -d . SEE/Externals.java

C:\aditya sharma java>javac Main.java

C:\aditya sharma java>java Main
Enter number of students: 1
Enter details for Student 1
Enter USN: 1bm22cs021
Enter Name: aditya sharma
Enter Semester: 3
Enter Internal marks for 5 subjects:
Enter marks for subject 1: 35
Enter marks for subject 2: 37
Enter marks for subject 3: 25
Enter marks for subject 4: 30
Enter marks for subject 5: 31
Enter External marks for 5 subjects:
Enter marks for subject 1: 95
Enter marks for subject 2: 85
Enter marks for subject 3: 90
Enter marks for subject 4: 80
Enter marks for subject 5: 96
USN: 1bm22cs021
Name: aditya sharma
Semester: 3
Internal Marks:
Subject 1: 35
Subject 2: 37
Subject 3: 25
Subject 4: 30
Subject 5: 31
Final Marks (CIE + SEE):
Subject 1: 130
Subject 2: 122
Subject 3: 115
Subject 4: 110
Subject 5: 127
```

<u>Program 7</u>

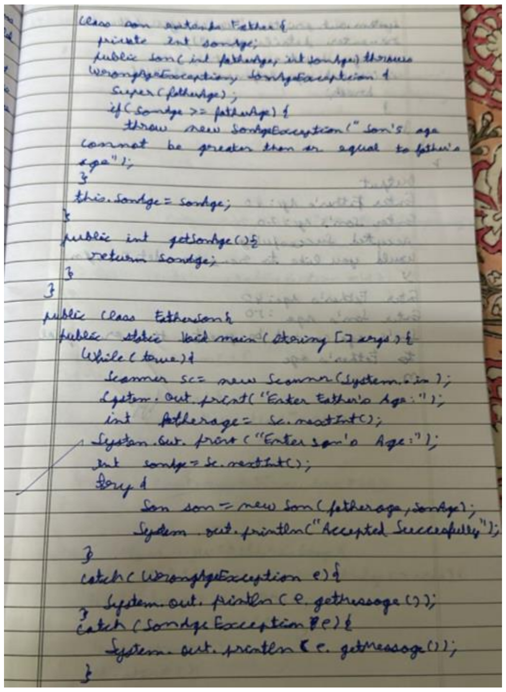
Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age=father's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

Algorithm:

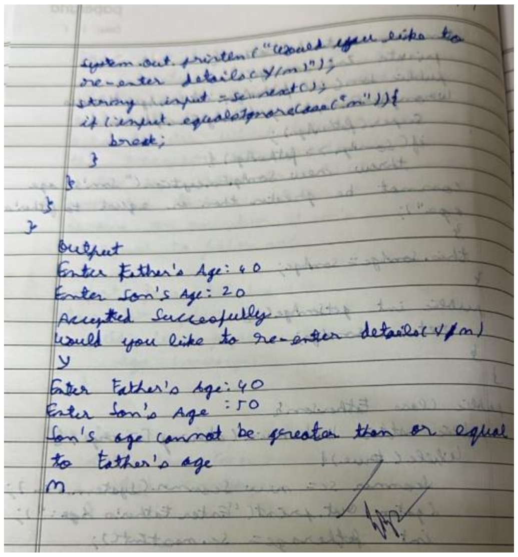


WAP that demonstrates handling of exceptions in inheritance tree create a base class called "Father" and a derived class called "Son" which extends the base class. In Father's class implement a constructor which takes the age and throws the exception WrongAge() when the input age is less than zero. In son's class implement a constructor that uses Father and Son's age and throws an exception if son's age is greater than or equal to fathers age

```
import java.util.Scanner;
class WrongAgeException extends Exception{
  public WrongAgeException (string message){
      super(message);
  }
}
class SonAgeException extends Exception{
    public SonAgeException (string message){
       super(message);
    }
}
class Father{
  private int age;
  public Father(int age) throws WrongAgeException{
     if (age <0) {
        throw new WrongAgeException("WrongAge");
     }
    this.age=age;
  }
public int getAge(){
   return age;
 }
}
```



```
class son extends Father{
  private int sonAge;
  public son( int fatherAge, int sonAge) throws
  WrongAgeException, SonAgeException {
     Super(fatherAge);
     if(sonAge >= fatherAge) {
       throw new SonAgeException(" Son's age
cannot be greater than or equal to father's
age");
  }
 this.SonAge= sonAge;
}
public int getSonAge(){
  return sonAge;
 }
}
public class FatherSon{
public static void main(String [] args){
   While(true){
      Scanner sc= new Scanner(System.in);
      System.out.print("Enter Father's Age:");
      int fatherage= sc.nextInt();
     System.out.print ("Enter son's Age:");
     int sonAge= sc.nextInt();
     try{
         Son son = new Son(fatherage,SonAge);
         System.out.println("Accepted Succesfully");
   }
   catch(WrongAgeException e){
     System.out.println(e.getMessage());
   }
   catch(SonAgeException e){
     System.out.println(e.getMessage());
   }
```

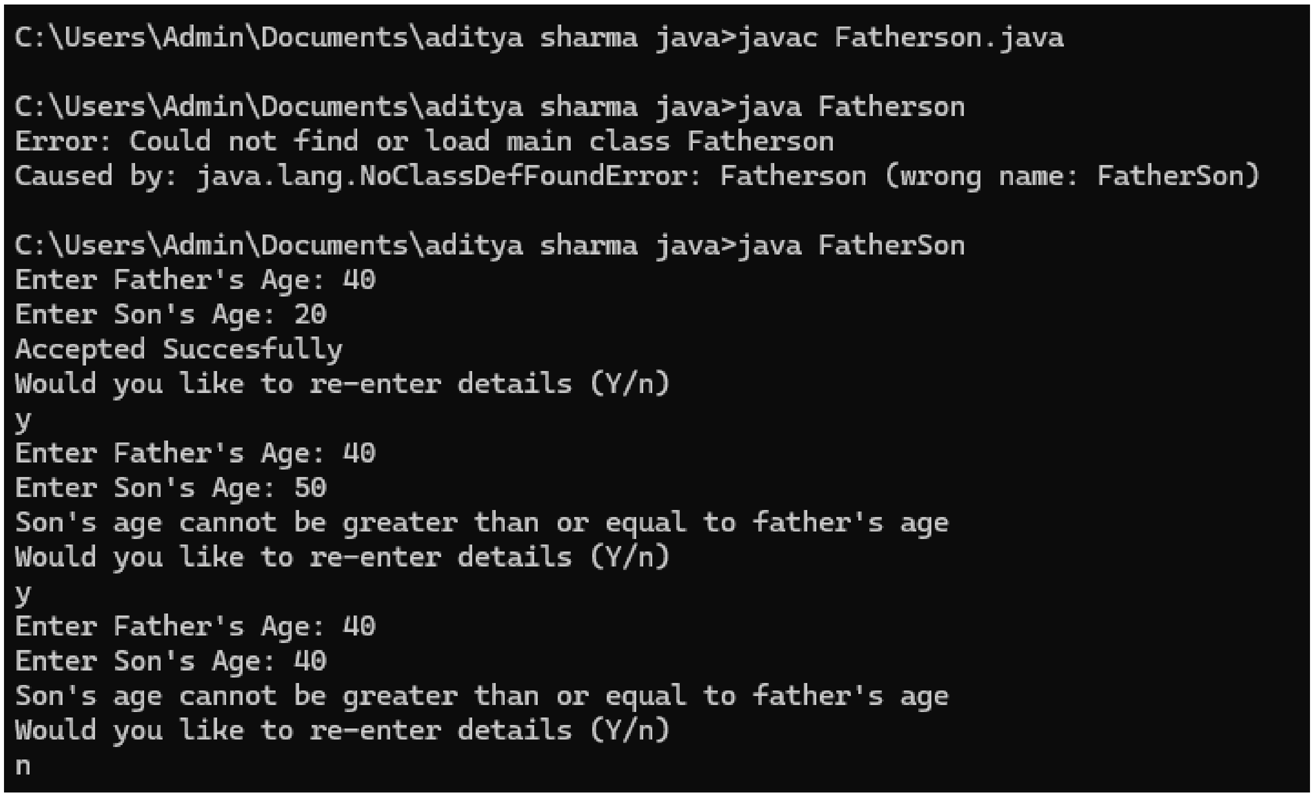


```
system.out.println("would you like to
re-enter details(y/n)");
string input =sc.next();
if(!input.equalsIgnoreCase("n")){
   break;
 }
}
}
}
Output
Enter Father's Age: 40
Enter Son's Age: 20
Accepted Successfully
Would you like to re-enter details(y/n)
y
Enter Father's Age: 40
Enter Son's Age :50
Son's age cannot be greater than or equal
to Father's age
n
```

Code:

```
import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

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

class Father {
    private int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Wrong age");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException, SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException("Son's age cannot be greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}
public class FatherSon{
    public static void main(String[] args) {
        while(true){
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter Father's Age: ");
            int fatherAge = sc.nextInt();
            System.out.print("Enter Son's Age: ");
            int sonAge = sc.nextInt();
            try {
                Son son = new Son(fatherAge, sonAge);
                System.out.println("Accepted Successfully");
            }
            catch (WrongAgeException e) {
                System.out.println(e.getMessage());
            }
            catch (SonAgeException e) {
                System.out.println(e.getMessage());
            }
            System.out.println("Would you like to re-enter details (Y/n)");
            String input = sc.next();
            if (input.equalsIgnoreCase("n")) {
                break;
            }
        }
    }
}
```

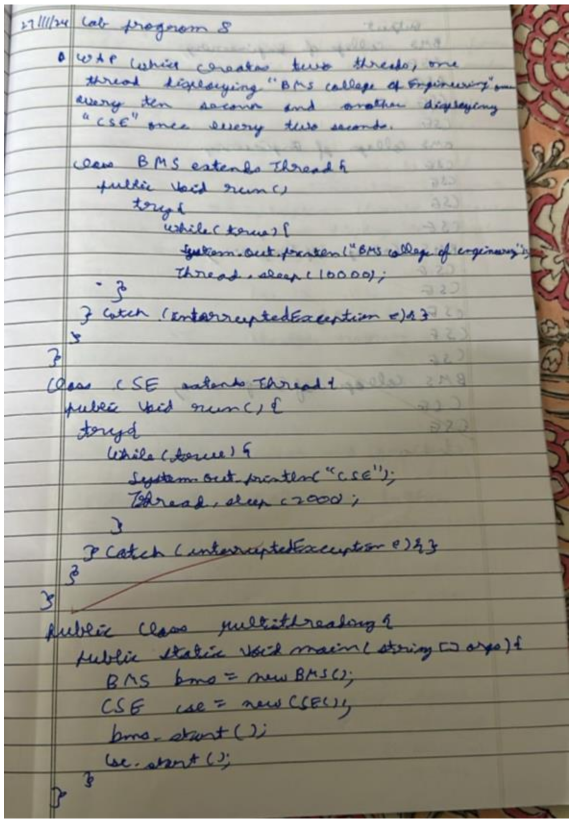
Output:



```
C:\Users\Admin\Documents\aditya sharma java>javac Fatherson.java

C:\Users\Admin\Documents\aditya sharma java>java Fatherson
Error: Could not find or load main class Fatherson
Caused by: java.lang.NoClassDefFoundError: Fatherson (wrong name: FatherSon)

C:\Users\Admin\Documents\aditya sharma java>java FatherSon
Enter Father's Age: 40
Enter Son's Age: 20
Accepted Succesfully
Would you like to re-enter details (Y/n)
y
Enter Father's Age: 40
Enter Son's Age: 50
Son's age cannot be greater than or equal to father's age
Would you like to re-enter details (Y/n)
y
Enter Father's Age: 40
Enter Son's Age: 40
Son's age cannot be greater than or equal to father's age
Would you like to re-enter details (Y/n)
n
```

<u>Program 8</u>
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.

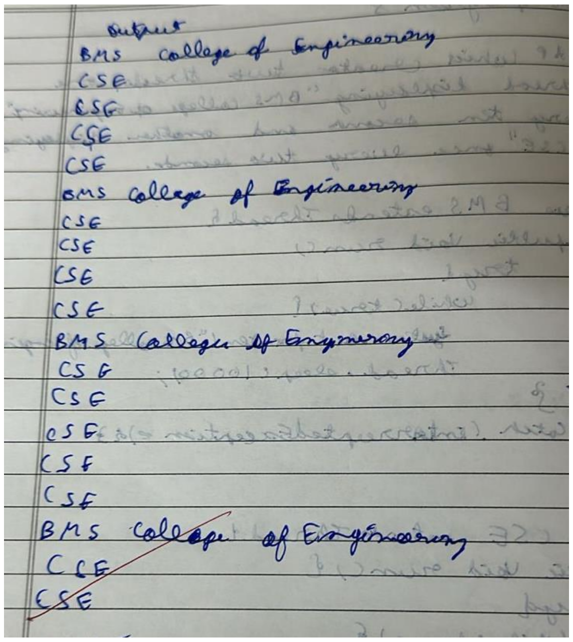
Algorithm:



27/11/24 Lab program 8

Q WAP which creates two threads, one thread displaying "BMS college of Engineering" once every ten second and another displaying "CSE" once every two seconds.

```
class BMS extends Thread {
  public void run()
     try{
        while(true){
           System.out.println("BMS college of engineering");
           Thread.sleep(10000);
     }
   } catch (InterruptedException e){}
  }
}
class CSE extends Thread{
 public void run(){
  try{
    while(true){
      System.out.println("CSE");
      Thread.sleep(2000);
    }
  } catch (InterruptedException e){}
 }
}
public class multithreading{
  public static void main(String[] args){
    BMS bms = new BMS();
    CSE cse = new CSE();
    bms.start();
    cse.start();
  }
}
```



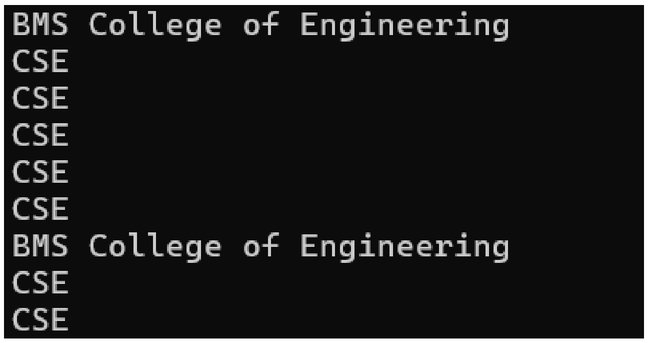
Output

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

Code:

```
class BMS extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000); // Sleep for 10 seconds
            }
        }catch (InterruptedException e) {}
    }
}
class CSE extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000); // Sleep for 2 seconds
            }
        }catch (InterruptedException e) {}
    }
}

public class Multithreading{
    public static void main(String[] args) {
        BMS bms = new BMS();
        CSE cse = new CSE();
        bms.start();
        cse.start();
    }
}
```

Output:

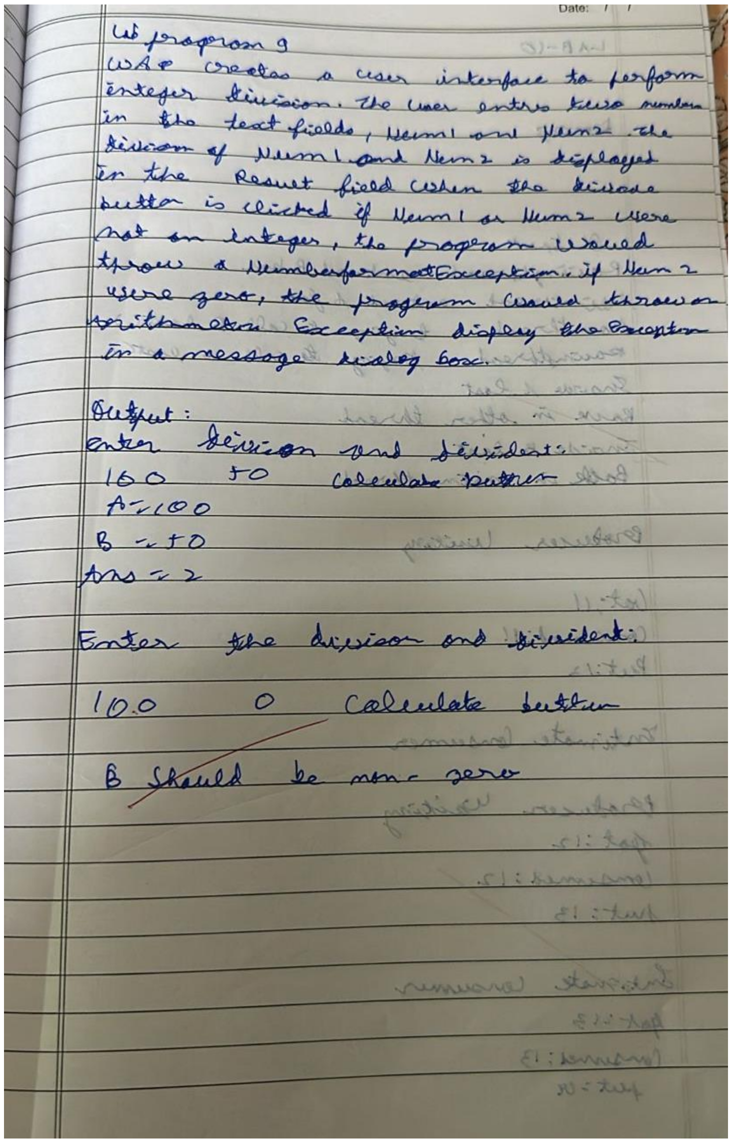


```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
```

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:



Date: / /

Lab program 9
WAP creates a user interface to perform integer division. The user entres two number 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 Num 2 were zero, the program would throw an Arithmetic Exception display the Exception in a message dialog box.

Output:
enter division and dividend:
100 50 calculate button
A = 100
B = 50
Ans = 2

Enter the divison and dividend:

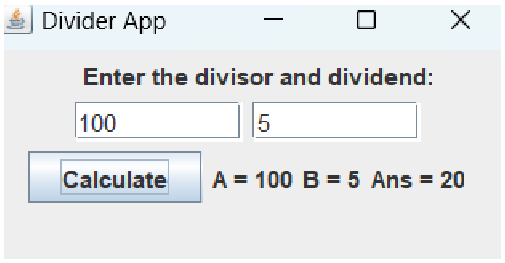
10.0 0 Calculate button

B Should be non - zero

Code:

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

class SwingDemo {
    SwingDemo() {
        // Create JFrame container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // Terminate the application when the frame is closed
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // Text label for input instructions
        JLabel jlab = new JLabel("Enter the divisor and dividend:");

        // Add text fields for the two numbers (dividend and divisor)
        JTextField ajtf = new JTextField(8); // Dividend input field
        JTextField bjtf = new JTextField(8); // Divisor input field

        // Button to trigger calculation
        JButton button = new JButton("Calculate");

        // Labels to display errors, inputs, and the result
        JLabel err = new JLabel();   // Error message label
        JLabel alab = new JLabel();  // Label for dividend value
        JLabel blab = new JLabel();  // Label for divisor value
        JLabel anslab = new JLabel(); // Label for the result (answer)

        // Add components to the JFrame
        jfrm.add(err);    // Error label
        jfrm.add(jlab);   // Instruction label
        jfrm.add(ajtf);   // Dividend input field
        jfrm.add(bjtf);   // Divisor input field
        jfrm.add(button);  // Calculate button
        jfrm.add(alab);   // Display dividend value
        jfrm.add(blab);   // Display divisor value
        jfrm.add(anslab); // Display the result

        // ActionListener for the "Calculate" button
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                try {
                    // Parse input values as integers
                    int a = Integer.parseInt(ajtf.getText());
                    int b = Integer.parseInt(bjtf.getText());

                    // Perform division and display the result
                    int ans = a / b;
                    alab.setText("A = " + a);
                    blab.setText("B = " + b);
                    anslab.setText("Ans = " + ans);
                    err.setText(""); // Clear error message if no exception

                } catch (NumberFormatException e) {
                    // Catch invalid integer inputs (non-numeric)
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("Enter Only Integers!");
                } catch (ArithmeticException e) {
                    // Catch division by zero
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("B should be NON-zero!");
                }
            }
        });

        // Make the frame visible
        jfrm.setVisible(true);
    }

    public static void main(String args[]) {
        // Create frame on event dispatching thread
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new SwingDemo(); // Initialize the SwingDemo frame
            }
        });
    }
}
```

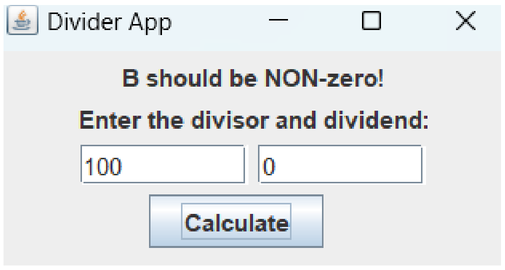
Output:



Enter the divisor and dividend:

100 5

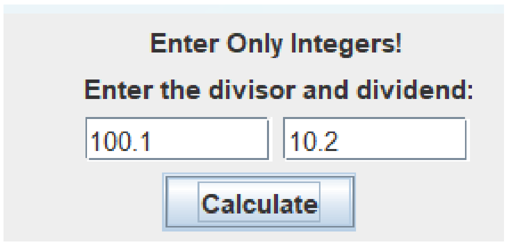
A = 100 B = 5 Ans = 20



B should be NON-zero!

Enter the divisor and dividend:

100 0



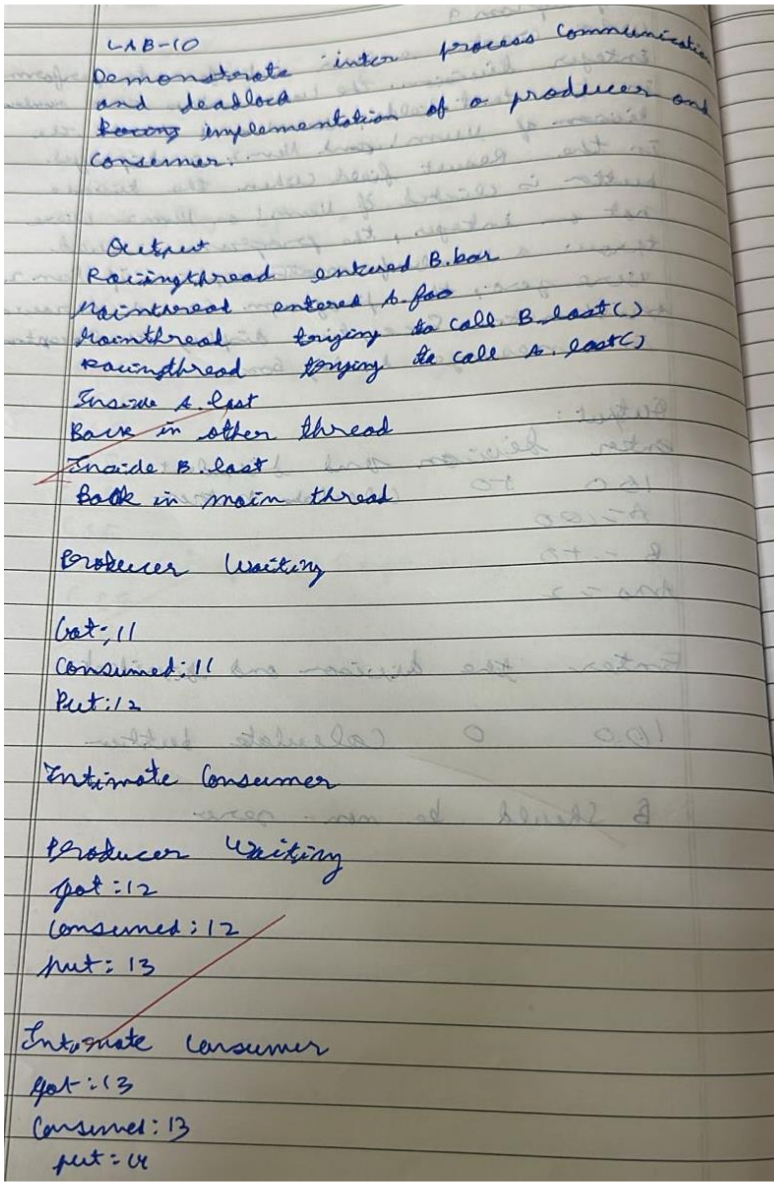
Enter Only Integers!

Enter the divisor and dividend:

100.1 10.2

Program 10
Demonstrate Inter process Communication and deadlock.

Algorithm:



LAB-10
Demonstrate inter process Communication and deadlock ~~Racing~~ implementation of a producer and Consumer.

Output
Racingthread entered B.bar
Mainthread entered A.foo
Mainthread trying to call B.last()
Racingthread trying to call A.last()
Inside A.last
Back in other thread
Inside B.last
Back in main thread

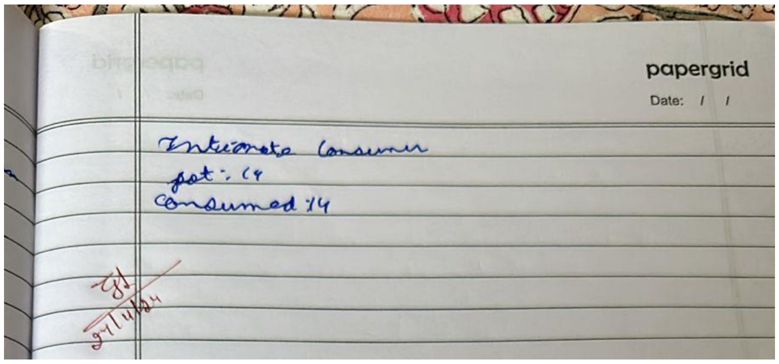
Producer Waiting

Got: 11
Consumed: 11
Put: 12

Intimate Consumer

Producer Waiting
got: 12
Consumed: 12
put: 13

Intimate Consumer
got: 13
Consumed: 13
put: 14



papergrid
Date: / /

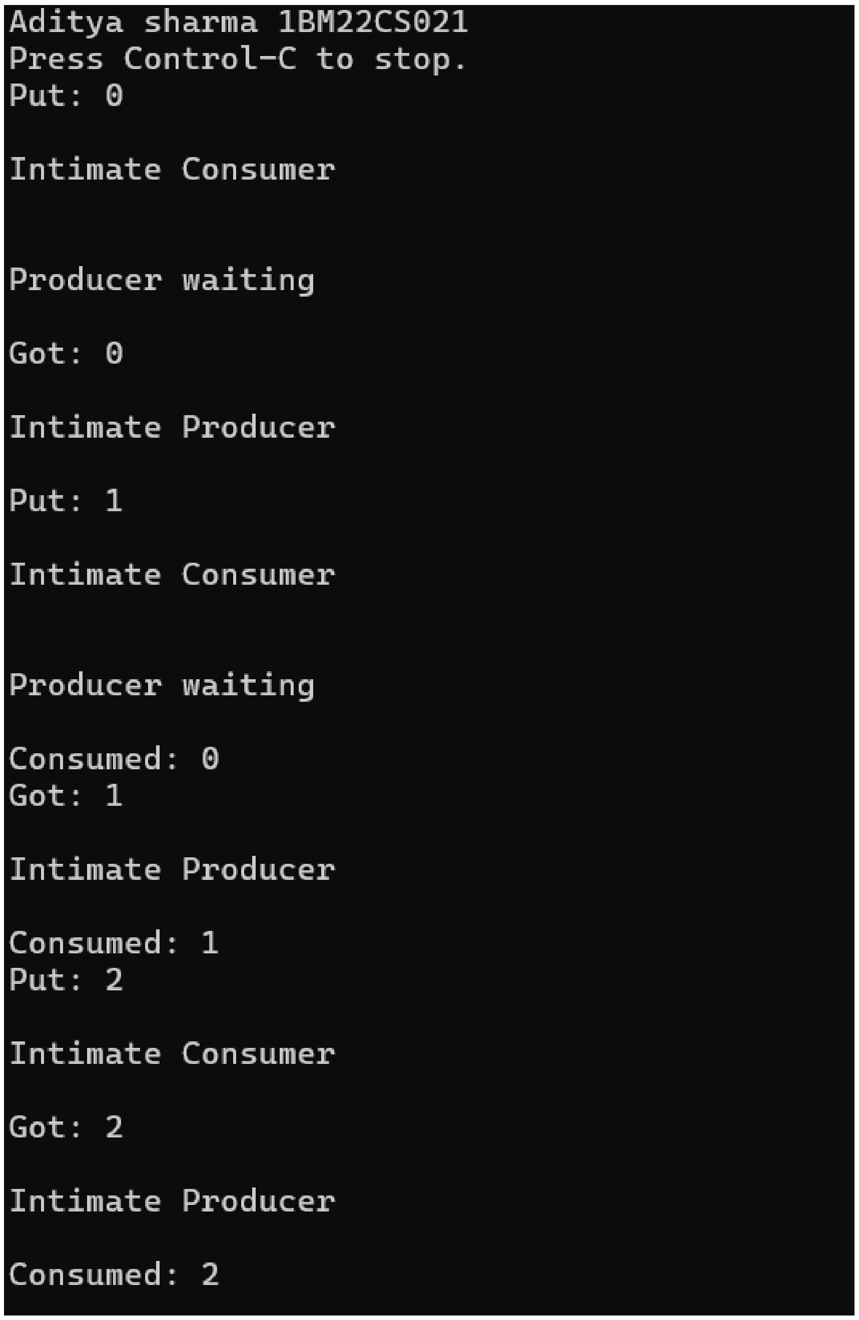
Intimate Consumer
got: 14
Consumed: 14

[illegible]
27/11/24

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 < 3) {
            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 < 3) {
            int r = q.get();
            System.out.println("Consumed: " + r);
            i++;
        }
    }
}

class PCFixed {
    public static void main(String args[]) {
        System.out.println("Aditya sharma 1BM22CS021");
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}
```

Output:



```
Aditya sharma 1BM22CS021
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

Got: 2

Intimate Producer

Consumed: 2
```