

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



**LAB REPORT**

**on**

**Object Oriented Java Programming**

**(1BM23CS174)**

*Submitted by*

**Mahamad Aziz Ansari**

*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 (1BM23CS174)” carried out by **StudentName (Mahamad Aziz Ansari)**, 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 (1BM23CS174) work prescribed for the said degree.

Lab faculty Incharge Name : <b>Srushhti C S</b>	Dr. Jyothi S Nayak
Assistant Professor	Professor & HOD
Department of CSE, BMSCE	Department of CSE, BMSCE

## Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	30/9/24	Quadratic Equation	4-9
2	6/10/24	Student SGPA	9-17
3	14/10/24	Book Details	17-23
4	21/10/24	Area of Shapes	23-28
5	28/10/24	Bank saving	29-39
6	11/11/24	Packages	39-46
7	28/11/24	Exceptional Handling	46-51
8	28/11/24	Threads	51-55
9	28/11/24	Open ended question (Division App )	55-61
10	30/11/21	Open ended Exercise (Inter process communication and deadlock )	61-71

Github Link:

<https://github.com/Mahamadazizansari/java-lab-Programes->

## **Program 1**

Implement Quadratic Equation

Algorithm:

Date : \_\_\_\_\_  
Page No. : \_\_\_\_\_

# Quadratic equation.

```

import java.util.*;
class Quadratic {
    double r1, r2, disc;
    int a, b, c;

    void get() {
        Scanner s = new Scanner (System.in);
        System.out.println ("enter the coefficients of
            quadratic eqn");
        a = s.nextInt();
        if (a==0) {
            System.out.println ("It should be a quadratic eqn.
                Enter value again");
            a = s.nextInt();
        }
        b = s.nextInt();
        c = s.nextInt();
    }

    void compute () {
        disc = (b*b)-(4*a*c);
        if (disc==0)
            r1 = (-b)/(2*a);
        else
            r1 = (-b + Math.sqrt(disc))/(2*a);
        r2 = r1;
        System.out.println ("roots of eqn are "
            +r1 + " and " + r2);
    }
}

```

```

else if (disc > 0)
{
    r1 = ((-b) + (Math.sqrt(disc))) / (double)(2.0 * a);
    r2 = ((-b) - (math.sqrt(disc))) / (double)(2.0 * a);
    System.out.println("roots of eqn are " + r1 + " and "
                        + r2);
}
else
{
    System.out.println("there are no real soln");
}
}

class Quadratic
{
    public static void main (String args[])
    {
        Quadratic q = new Quadratic();
        q.get ("");
        q.compute();
        System.out.println("Name: Mahamad Aziz Ansari");
        System.out.println("USN: 1BM23CS174");
    }
}

```

### # Output

Enter coefficient of a : 3

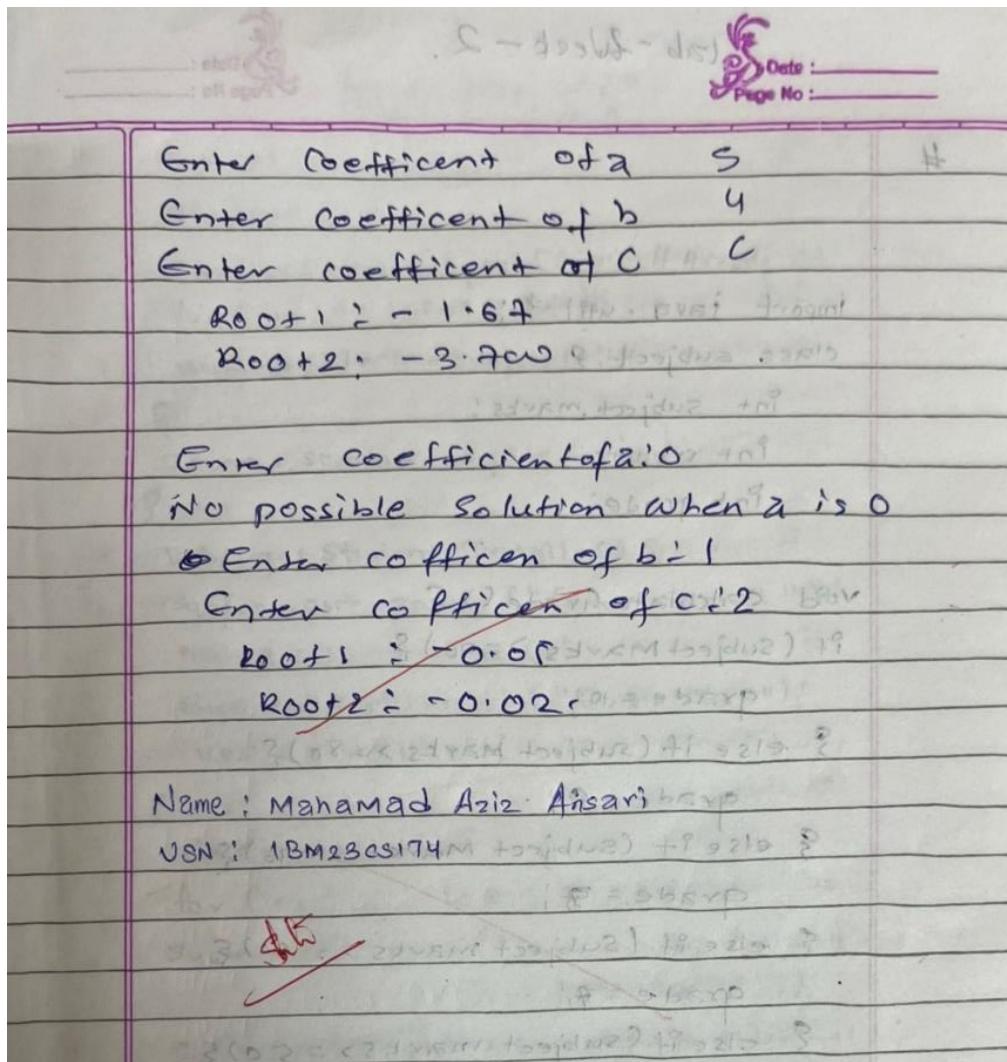
Enter coefficient of b : -6

Enter coefficient of c : 9

Roots are real and non imaginary i.e.

Root 1 : -1.0

Root 2 : 1.4142130623



```
import java.util.*;  
class Quadratic{  
double r1,r2,disc;  
int a,b,c;
```

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

```
System.out.println("enter the coefficients of Quadratic eqn");
```

```

a=s.nextInt();
if (a==0){
System.out.println("It should be a quadratic eqn. Enter values again");
a=s.nextInt();}

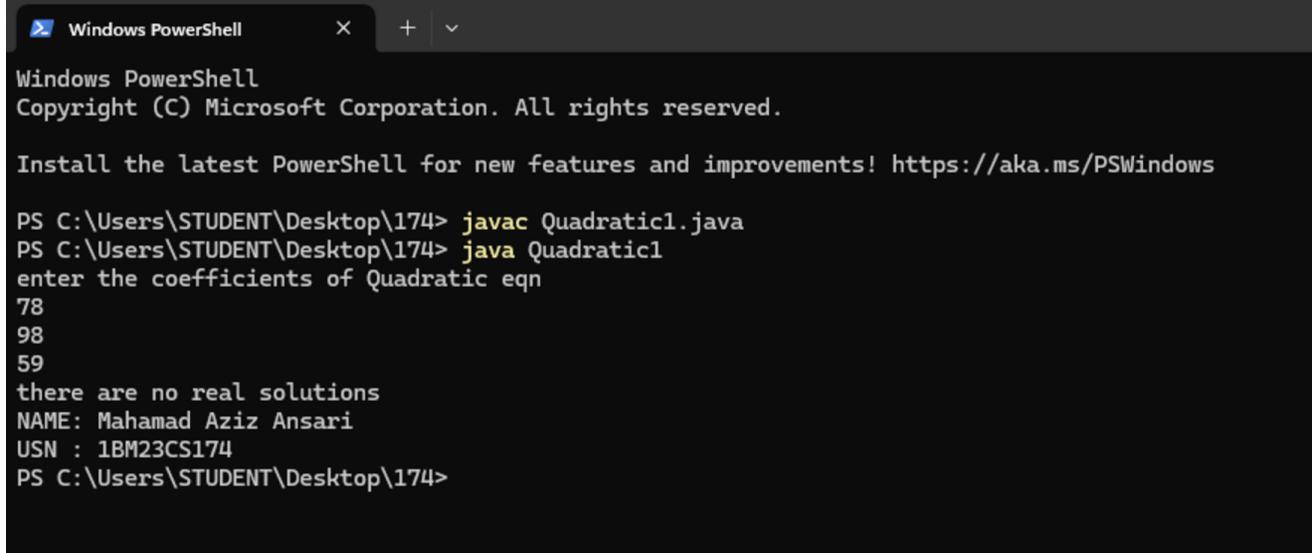
b=s.nextInt();
c=s.nextInt();
}
void compute()

{
disc=(b*b)-(4*a*c);
if (disc==0)
{
r1=(-b)/2*a;
r2=r1;
System.out.println("roots of eqn are "+r1+"and"+r2);
}
else if (disc>0)
{
r1=((-b)+ (Math.sqrt(disc)))/(double)(2.0*a);
r2=((-b)- (Math.sqrt(disc)))/(double)(2.0*a);
System.out.println("roots of eqn are " +r1+ "and" +r2);
}
else{
System.out.println("there are no real solutions");
}
}
}

class Quadratic1{
public static void main(String args[]){
Quadratic q=new Quadratic();
q.get();
q.compute();
System.out.println("NAME: Mahamad Aziz Ansari");
}

```

```
System.out.println("USN : 1BM23CS174");
}}
```



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the command PS C:\Users\STUDENT\Desktop\174> java Quadratic1 and its output. The output includes prompts for coefficients, the message "there are no real solutions", and the student's name and USN.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\STUDENT\Desktop\174> javac Quadratic1.java
PS C:\Users\STUDENT\Desktop\174> java Quadratic1
enter the coefficients of Quadratic eqn
78
98
59
there are no real solutions
NAME: Mahamad Aziz Ansari
USN : 1BM23CS174
PS C:\Users\STUDENT\Desktop\174>
```

## Program 2: STUDENT SGPA

Algorithm:

## Lab - Week - 2.

 Date : \_\_\_\_\_  
 Page No. : \_\_\_\_\_

#

```

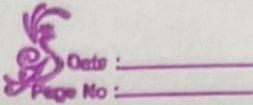
    public class Subject {
        int subject_marks;
        int credits;
        String grade;

        void calculateGrade() {
            if (subject_marks >= 90) {
                grade = "A";
            } else if (subject_marks >= 80) {
                grade = "B";
            } else if (subject_marks >= 70) {
                grade = "C";
            } else if (subject_marks >= 60) {
                grade = "D";
            } else {
                grade = "F"; // Fail
            }
        }

        class Student {
            String name;
            String USN;
            double CGPA;
            Scanner s;
        }
    }

```

import. java.util.\*;  
class Subject {



Subject subject [ ];

Student () {

Subject = new Subject [ ] ; // Array of 9 sub  
for (int i = 0 ; i < 8 ; i++) {

Subject[i] = new Subject () ; //

}

s = new Scanner (System.in);

{

Void get Student Details () {

System.out.print ("Enter Name : ");

Name = s.nextLine ();

System.out.print ("Enter USN : ");

USN = s.nextLine ();

}

Void get marks () {

for (int i = 0 ; i < 8 ; i++) {

System.out.print ("Enter marks for Subject " +

(i+1) + " : ");

Subject[i].subjectmarks = s.nextInt ();

System.out.print ("Enter credits for student " +

(i+1) + " : ");

Subject[i].credits = s.nextInt ();

Subject[i].calculate Grade ();

else if (subject[i].subjectmarks > 100) {

System.out.println ("Invalid marks. Marks  
should not exceed 100.");

Subject[i].subjectmarks = 100;

? else if (subject[i].subjectmarks < 0) {

System.out.println ("Invalid marks. Marks  
should not be negative.");

Subject[i].subjectMarks = 0;

```

/*
 * This program is designed to calculate the SGPA of a student based on their marks in 8 subjects.
 */

class Student {
    double totalCredits = 0;
    double totalGradePoints = 0;

    void computeSGPA() {
        for (int i = 0; i < 8; i++) {
            totalCredits += subject[i].credits;
            totalGradePoints += subject[i].credits *
                (subject[i].grade / 10);
        }
    }

    if (totalCredits > 0) {
        SGPA = totalGradePoints / totalCredits;
    } else {
        SGPA = 0; // Handle case with zero credits
    }
}

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

public static void main(String[] args) {
    Student s1 = new Student();
    s1.getStudentDetails();
    s1.getMarks();
    s1.computeSGPA();
    s1.displayResult();
}

```

Date: \_\_\_\_\_  
 Page No: \_\_\_\_\_

Enter USN: 174 (written down on a sheet) 18  
 Enter marks for subject 1: 86  
 Enter credit for subject 1: 2  
 Enter marks for Subject 2: 56  
 Enter credit for subject 2: 4 (d)  
 Enter marks for Subject 3: 99  
 Enter credit for Subject 3: 1  
 Enter marks for Subject 4: 85  
 Enter credit for subject 4: 3  
 Enter marks for Subject 5: 78  
 Enter credit for subject 5: 2  
 Enter marks for Subject 6: 66  
 Enter credit for subject 6: 4  
 Enter marks for Subject 7: 89  
 Enter ~~marks~~<sup>credit</sup> for subject 7: 3  
 Enter marks for Subject 8: 78  
 Enter credits for Subject 8: 2.  
 Name: Mahamah ~~22~~ ~~answering~~  
 USN: 174 (written down on a sheet)  
 SCRA: 7.42 (written down on a sheet)

student, vertices prints (short prints) good sheet  
 3 (written down on a sheet)  
 : SMDA = SMDA - 20  
 : vertices = vertices - 20  
 : SCRA = SCRA - 20  
 : SCRA = SCRA - 20

3 (3 printed prints sitting  
 written to "vertices" + "SCRA" ; "SCRA" written  
 "vertices" ; written to "vertices" + "SCRA" ; "SCRA" +

## Code:

```
import java.util.*;  
  
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; // Fail  
        }  
    }  
}  
}
```

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

```
Student() {
```

```

subject = new Subject[8]; // Array of 9 subjects
for (int i = 0; i < 8; i++) {
        subject[i] = new Subject(); // Create an array of Subject objects
}
s = new Scanner(System.in);
}

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

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

    subject[i].calculateGrade();

    if (subject[i].subjectMarks > 100) {
        System.out.println("Invalid marks. Marks should not exceed
100.");
            subject[i].subjectMarks = 100;
    } else if (subject[i].subjectMarks < 0) {
            System.out.println("Invalid marks. Marks should not be
negative.");
            subject[i].subjectMarks = 0;
    }
}

```

```

}

void computeSGPA() {
    double totalCredits = 0;
    double totalGradePoints = 0;

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

    if (totalCredits > 0) {
        SGPA = totalGradePoints / totalCredits;
    } else {
        SGPA = 0; // Handle case with zero credits
    }
}

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

public static void main(String[] args) {
    Student s1 = new Student();
    s1.getStudentDetails();
    s1.getMarks();
    s1.computeSGPA();
    s1.displayResults();
}
}

```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\STUDENT\Desktop\174> javac Student.java
PS C:\Users\STUDENT\Desktop\174> java Student
Enter Name: mahamad aziz ansari
Enter USN: 174
Enter marks for Subject 1: 86
Enter credits for Subject 1: 2
Enter marks for Subject 2: 56
Enter credits for Subject 2: 4
Enter marks for Subject 3: 89
Enter credits for Subject 3: 1
Enter marks for Subject 4: 85
Enter credits for Subject 4: 3
Enter marks for Subject 5: 78
Enter credits for Subject 5: 2
Enter marks for Subject 6: 66
Enter credits for Subject 6: 4
Enter marks for Subject 7: 59
Enter credits for Subject 7: 3
Enter marks for Subject 8: 78
Enter credits for Subject 8: 2
Name: mahamad aziz ansari
USN: 174
SGPA:7.428571428571429
PS C:\Users\STUDENT\Desktop\174>
```

## Program 3

### *Books Details*

Algorithm:



Q.A

Create a class Book which contains four members : name, author, Price, num\_pages. Include a constructor to set the value for the members. Include a method to set and get the details of the object. Include `toString()` method that could display the complete details of the book. Develop a Java program to create n book object.

\* Explore `toString()` method using `in.java`

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
```

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

```
public Book (String name, String author, double  

    price, int numPages) {  

    this.name = name;  

    this.author = author;  

    this.price = price;  

    this.numPages = numPages;
```

```
public String toString() {  

    return "Name : " + name + ", Author : " + author  

        + ", Price : " + price + ", Pages : " + numPages;
```

Date : \_\_\_\_\_  
Page No. : \_\_\_\_\_

```

8
{
    System.out.println("Enter book name:");
    String name = scanner.nextLine();
    System.out.print("Enter author name: ");
    String author = scanner.nextLine();
    System.out.print("Enter price: ");
    int price = scanner.nextInt();
    scanner.nextLine();

    Book book = new Book(name, author, price,
        numPages);
    books.add(book);
}

System.out.println("Details of the
books:");
for (Book book : books) {
    System.out.println(book);
}

System.out.println("mohamed aziz
ansari\nIBM23city");
Scanner.close();
}

```

Enter book name : harry

Enter author name : potter

Enter price : 789

Enter page num = 200

(will print) name? price? page?

(will print) Enter book name : king

Enter author name : kong

Enter price : 560

Enter page num = 49

(will print) name? price? page?

(will print) Enter book name : man

Enter author name : made

Enter price : 1590

Enter page num : 89

(will print) name? price? page?

Details.

Name: harry Author: potter price: 789, page:

Name: king Author: kong price: 560, page: 49

Name: man Author: made price: 1590, page: 89

will print "Harry" Author: potter price: 789

(will print)

Book name : king Author : kong price : 560

(will print) name? price? page?

Book name : man Author : made price : 1590

(will print) name? price? page?

(will print) name? price? page?

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

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

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

    @Override
    public String toString() {
        return "Name: " + name + ", Author: " + author + ", Price: " + price +
        ", Pages: " + numPages;
    }
}

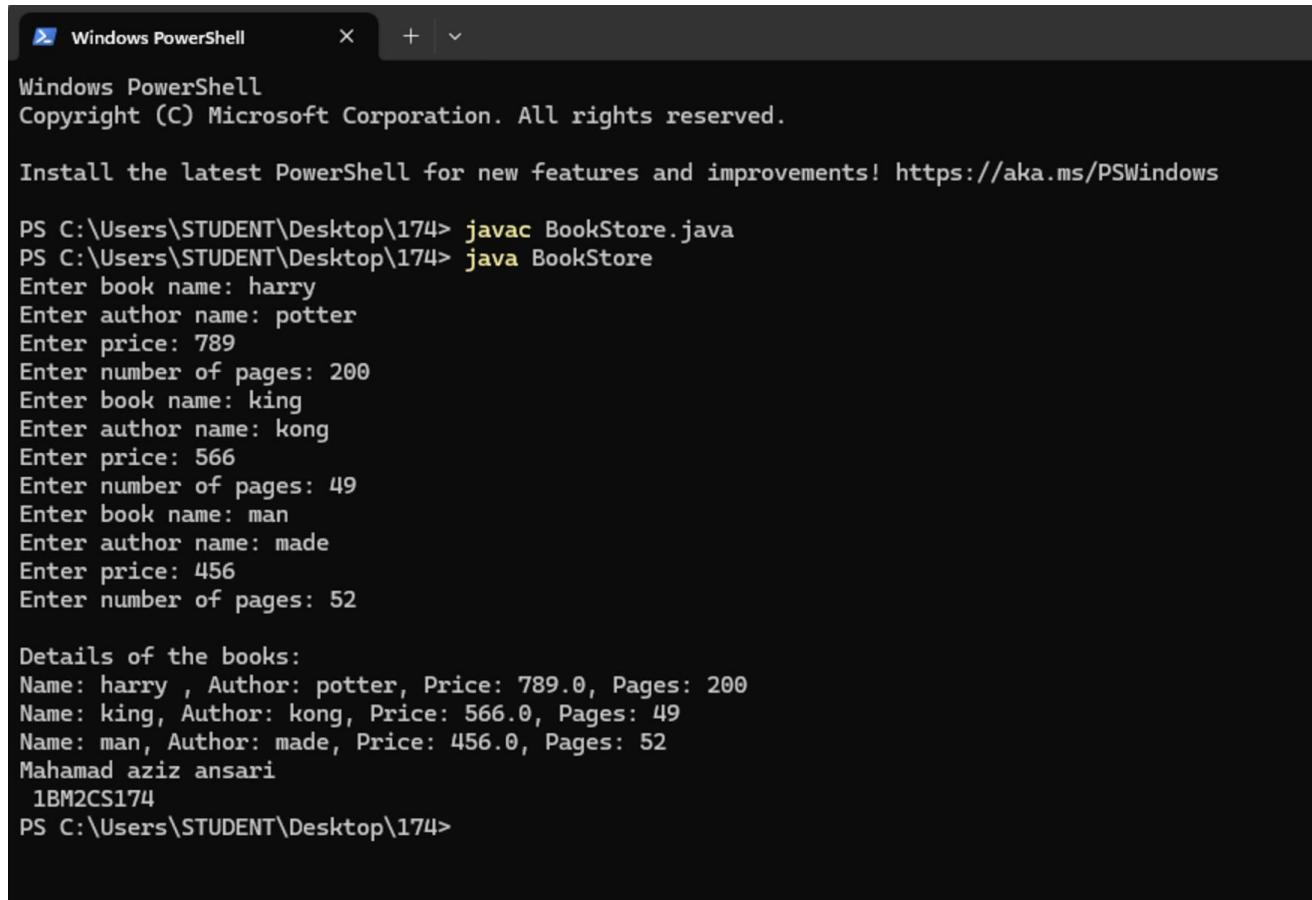
public class BookStore {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        List<Book> books = new ArrayList<>();

        for (int i = 0; i < 3; i++) {
            System.out.print("Enter book name: ");
            String name = scanner.nextLine();
            System.out.print("Enter author name: ");
            String author = scanner.nextLine();
        }
    }
}
```

```
System.out.print("Enter price: ");
int price = scanner.nextDouble();
System.out.print("Enter number of pages: ");
int numPages = scanner.nextInt();
scanner.nextLine();

Book book = new Book(name, author, price, numPages);
books.add(book);
}

System.out.println("\nDetails of the books:");
for (Book book : books) {
    System.out.println(book);
}
System.out.println("Mahamad aziz ansari \n 1BM2CS174");
scanner.close();
}
}
```



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the output of a Java application named "BookStore". The application prompts the user to enter details for three books: "harry" (author: potter, price: 789, pages: 200), "king" (author: kong, price: 566, pages: 49), and "man" (author: made, price: 456, pages: 52). It also displays the name "Mahamad aziz ansari". The command "javac BookStore.java" is run first, followed by "java BookStore". The entire session is displayed in a black terminal window.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\STUDENT\Desktop\174> javac BookStore.java
PS C:\Users\STUDENT\Desktop\174> java BookStore
Enter book name: harry
Enter author name: potter
Enter price: 789
Enter number of pages: 200
Enter book name: king
Enter author name: kong
Enter price: 566
Enter number of pages: 49
Enter book name: man
Enter author name: made
Enter price: 456
Enter number of pages: 52

Details of the books:
Name: harry , Author: potter, Price: 789.0, Pages: 200
Name: king, Author: kong, Price: 566.0, Pages: 49
Name: man, Author: made, Price: 456.0, Pages: 52
Mahamad aziz ansari
1BM2CS174
PS C:\Users\STUDENT\Desktop\174>
```

Program :4

Area of Shapes

Algorithm:

### Lab 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.

```

import java.util.Scanner;
abstract class Shape {
    abstract void printArea();
}

class Rectangle extends Shape {
    int length, width;
    Rectangle() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter length of rec:");
        this.length = scanner.nextInt();
        System.out.print("Enter the width of rec:");
        this.width = scanner.nextInt();
    }
    void printArea() {
        int area = length * width;
        System.out.println("Rectangle Area :" + area);
    }
}

class Triangle extends Shape {
    int base, height;
}

```

### Triangle()

```

Scanner scanner = new Scanner (System.in);
System.out.print ("Enter base of triangle: ");
this.base = scanner.nextInt ();
System.out.print ("Enter height of triangle: ");
this.height = scanner.nextInt ();
}

void PrintArea () {
    double area = 0.5 * base * height;
    System.out.println ("TriangleArea: " + area);
}

```

### class Circle extends Shape

```

int radius;
Circle () {
}

```

```

Scanner scanner = new Scanner (System.in);
System.out.print ("Enter radius of circle: ");
this.radius = scanner.nextInt ();
}

```

### void printArea ()

```

double area = Math.PI * radius * radius;
System.out.println ("CircleArea: " + area);
}

```

### Public class main

```

Public static void main (String [] args) {
}

```

```

Scanner scanner = new Scanner (System.in);
}

```

```

System.out.println ("Choose a shape: 1. Rectangle");
}

```

```

1.2. Triangle 1.3. Circle");
}

```

```

int choice = scanner.nextInt ();
}

```

Switch (choice) {

case 1:

shape rectangle = new Rectangle();

rectangle. print Area();

break

case 2:

shape triangle = new Triangle();

triangle. print Area();

break

case 3 :

shape circle = new Circle();

circle. print Area();

break

default:

System.out.println ("Invalid choice");

Output:

Rectangle :

Enter dimension 1 : 78

Enter dimension 2 : 78

Area of rectangle : 6048

Triangle :

Enter dimension 1 : 87

Enter dimension 2 : 98

Area of Triangle : 4263.0

Circle :

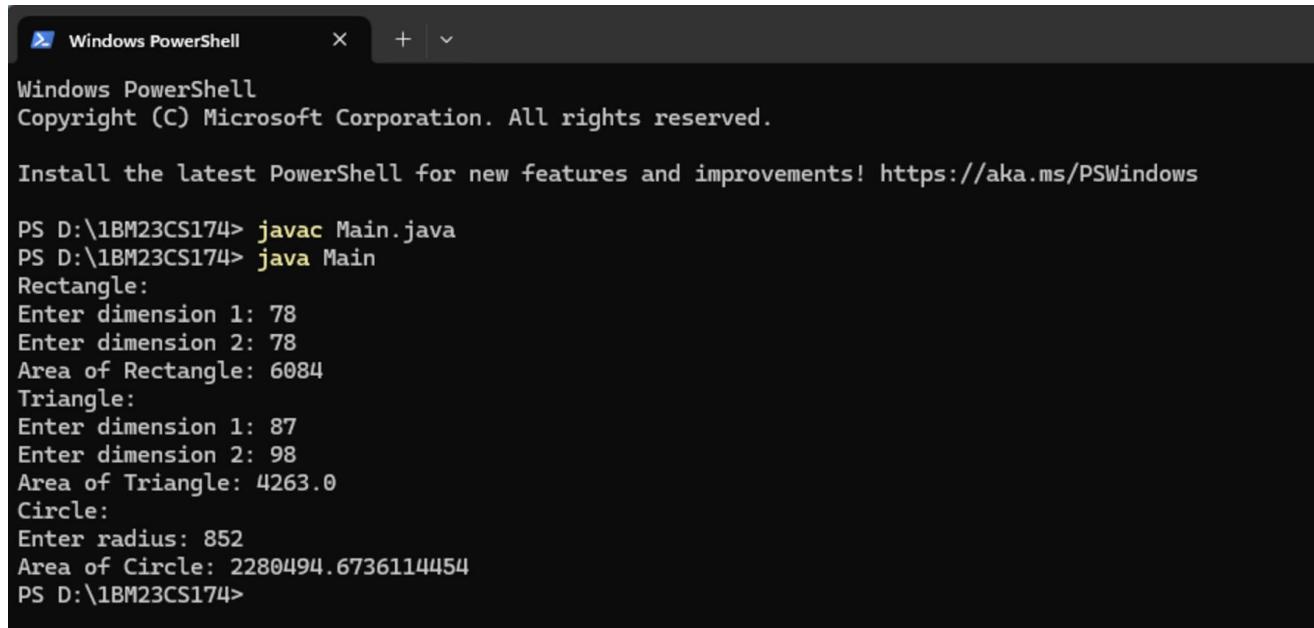
Enter radius : 852

Area of circle : 2280494.6736

Code :

```
import java.util.Scanner;
class InputScanner {
protected Scanner scanner = new Scanner(System.in);
}
abstract class Shape extends InputScanner {
protected int dimension1;
protected int dimension2;
public abstract void printArea();
public void input() {
System.out.print("Enter dimension 1: ");
dimension1 = scanner.nextInt();
System.out.print("Enter dimension 2: ");
dimension2 = scanner.nextInt();
}
}
class Rectangle extends Shape {
public void printArea() {
int area = dimension1 * dimension2;
System.out.println("Area of Rectangle: " + area);
}
}
class Triangle extends Shape {
public void printArea() {
double area = 0.5 * dimension1 * dimension2;
System.out.println("Area of Triangle: " + area);
}
}
class Circle extends Shape {
public void printArea() {
double area = Math.PI * Math.pow(dimension1, 2);
System.out.println("Area of Circle: " + area);
}
}
```

```
public class Main {  
    public static void main(String[] args) {  
        InputScanner inputScanner = new InputScanner();  
        System.out.println("Rectangle:");  
        Rectangle rectangle = new Rectangle();  
        rectangle.input();  
        rectangle.printArea();  
        System.out.println("Triangle:");  
        Triangle triangle = new Triangle();  
        triangle.input();  
        triangle.printArea();  
        System.out.println("Circle:");  
        Circle circle = new Circle();  
        System.out.print("Enter radius: ");  
        circle.dimension1 = inputScanner.scanner.nextInt();  
        circle.printArea();  
        inputScanner.scanner.close();  
    }  
}
```



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the execution of a Java application named "Main". The application prompts for user input for a rectangle, a triangle, and a circle, and then prints their respective areas. The PowerShell window has a dark theme and is running on a Windows operating system.

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS D:\1BM23CS174> javac Main.java  
PS D:\1BM23CS174> java Main  
Rectangle:  
Enter dimension 1: 78  
Enter dimension 2: 78  
Area of Rectangle: 6084  
Triangle:  
Enter dimension 1: 87  
Enter dimension 2: 98  
Area of Triangle: 4263.0  
Circle:  
Enter radius: 852  
Area of Circle: 2280494.6736114454  
PS D:\1BM23CS174>
```

Program :5

Bank Saving

Algorithm:

Lab: 5.

Develop a Java program to create a class Bank that maintains two kinds of account for its customer, one called saving account and the other one called current account. The saving account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holder 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 class Cur-Acc and Sav-Acc to make them more specific to their requirement. Include the necessary methods in order to achieve the following task

- a. Accept deposit from customer and update balance
- b. Display the balance
- c. Compute and deposit interest
- d. Permit withdrawal and update the balance
- e. Check the minimum balance, impose penalty if necessary and update the balance

```

import java.util.Scanner;
class Account {
    protected String customerName;
    protected String accountNumber;
    protected double balance;
    public Account (String customerName, String
                    accountNumber) {
    }

```

```

        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = 0.0;
    }

```

```

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

```

```

    class SavingsAccount extends Account {
        private double interestRate;
        public SavingsAccount (String customerName, String
                               accountNumber, double interestRate) {
            Super (Customer Name, account Number);
            this.interestRate = interestRate;
        }
    }

```

```

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

```

```

    public boolean withdraw (double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
        }
    }

```

```
System.out.println ("withdrawn: " + amount);
return true;
```

```
}
```

```
return false;
```

```
}
```

```
}
```

```
class CurrentAccount extends Account {
```

```
private static final double MIN_BALANCE = 1000;
```

```
private static final double SERVICE_CHARGE = 50.0;
```

```
public CurrentAccount (String customerName,
```

```
String accountNumber) {
```

```
super (customerName, accountNumber);
```

```
}
```

```
public boolean withdraw (double amt) {
```

```
if (amt > 0 && amt <= balance) {
```

```
balance -= amount;
```

```
if (balance < MIN_BALANCE) {
```

```
balance -= SERVICE_CHARGE;
```

```
System.out.println ("Service charge imposed.
```

```
New balance: " + balance);
```

```
}
```

```
System.out.println ("Withdrawn: " + amount)
```

```
return true;
```

```
}
```

```
return false; // return false if withdrawal
```

```
if (balance + amount < MIN_BALANCE)
```

```
}
```

```
public class Bank {
```

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

```
Scanner scanner = new Scanner (System.in);
```

```
System.out.print ("Enter customer name: ");
```

```
String name = scanner.nextLine();
```

```

System.out.print("Enter account number: ");
String accountNumber = scanner.nextLine();
System.out.print("choose acc type (1. Saving
                2. for current): ");
int choice = scanner.nextInt();
Account account = (choice == 1)
    ? new SavingsAccount(name, accountNumber, 5.0)
    : new CurrentAccount(name, accountNumber);
while (true) {
    System.out.println("1.menu | 2. Deposite | 3. Disp-
        play balance | 4. withdraw | 5. Compute
        Interest (saving only) | 6. Exit");
    System.out.print("Choose option: ");
    int option = scanner.nextInt();
    switch (option) {
        case 1:
            System.out.print("Enter amt to deposit: ");
            account.deposit(scanner.nextDouble());
            break;
        case 2:
            account.displayBalance();
            break;
        case 3:
            System.out.print("Enter withdraw: ");
            double withdrawAmt = scanner.nextDouble();
            if (account instanceof SavingAccount) {
                ((SavingAccount) account).withdraw(
                    withdrawAmt);
            } else {
                ((CurrentAccount) account).withdraw(
                    withdrawAmt);
            }
    }
}

```

break;

case case 4:

if (account instance of Saving Account) {  
((Saving Account) account). compute AND  
deposit Interest();

else {

System.out.println ("Interest computation not  
available for current Account");  
break;

case 5:

System.out.println ("Exiting--");

Scanner.close();

return;

default:

System.out.println ("Invalid ");

{

{

{

{

Outputs: Enter c.name = Mahamad aziz ansari

Enter ac.number = 0000000003

choose acc type (1. Saving 2. Current): 1

menu

1. Deposit 2. Display balance 3. withdraw 4. Compute Interest

choose option : 1

Enter deposit amt: 999999

Deposited amt : 9.9999

menu:

1. Deposit . 2. Display balance 3. withdraw 4. Compute Interest

choose an option 3.

Enter the withdraw amt : 333333

withdrawn : 3.333

Code :

```
import java.util.Scanner;
class Account {
    String customerName;
    String accountNumber;
    double balance;
    public Account(String customerName, String accountNumber) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = 0.0;
    }
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        }
    }
    public void displayBalance() {
        System.out.println("Current balance: " + balance);
    }
}
class SavingsAccount extends Account {
    private double interestRate;
    public SavingsAccount(String customerName, String accountNumber,
        double interestRate) {
        super(customerName, accountNumber);
        this.interestRate = interestRate;
    }
    public void computeAndDepositInterest() {
        double interest = balance * interestRate / 100;
        deposit(interest);
        System.out.println("Interest deposited: " + interest);
    }
    public boolean withdraw(double amount) {
```

```

if (amount > 0 && amount <= balance) {
balance -= amount;
System.out.println("Withdrawn: " + amount);
return true;
}
return false;
}

class CurrentAccount extends Account {
private static final double MIN_BALANCE = 1000.0;
private static final double SERVICE_CHARGE = 50.0;
public CurrentAccount(String customerName, String accountNumber) {
super(customerName, accountNumber);
}
public boolean withdraw(double amount) {
if (amount > 0 && amount <= balance) {
balance -= amount;
if (balance < MIN_BALANCE) {
balance -= SERVICE_CHARGE;
System.out.println("Service charge imposed. New balance: " + balance);
}
System.out.println("Withdrawn: " + amount);
return true;
}
return false;
}
}

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

System.out.print("Enter customer name: ");
String name = scanner.nextLine();

System.out.print("Enter account number: ");

```

```

String accountNumber = scanner.nextLine();

System.out.print("Choose account type (1 for Savings, 2 for Current): ");
int choice = scanner.nextInt();

Account account = (choice == 1)
    ? new SavingsAccount(name, accountNumber, 5.0) // 5% interest for
        simplicity
    : new CurrentAccount(name, accountNumber);

while (true) {
    System.out.println("\nMenu:\n1. Deposit\n2. Display Balance\n3.
        Withdraw\n4. Compute Interest (Savings only)\n5. Exit");
    System.out.print("Choose an option: ");
    int option = scanner.nextInt();
    switch (option) {
        case 1:
            System.out.print("Enter amount to deposit: ");
            account.deposit(scanner.nextDouble());
            break;
        case 2:
            account.displayBalance();
            break;
        case 3:
            System.out.print("Enter amount to withdraw: ");
            double withdrawAmount = scanner.nextDouble();
            if (account instanceof SavingsAccount) {
                ((SavingsAccount) account).withdraw(withdrawAmount);
            } else {
                ((CurrentAccount) account).withdraw(withdrawAmount);
            }
            break;
        case 4:
            if (account instanceof SavingsAccount) {
                ((SavingsAccount) account).computeAndDepositInterest();
            } else {

```

```
System.out.println("Interest computation not available for Current  
Account.");  
}  
break;  
case 5:  
System.out.println("Exiting...");  
scanner.close();  
return;  
default:  
System.out.println("Invalid option. Try again.");  
}  
}  
}  
}
```

```
Windows PowerShell X + 
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\STUDENT\Desktop\LAB 5> javac Bank.java
PS C:\Users\STUDENT\Desktop\LAB 5> java Bank
Enter customer name: Mahamad Aziz Ansari
Enter account number: 00000000000003
Choose account type (1 for Savings, 2 for Current): 1

Menu:
1. Deposit
2. Display Balance
3. Withdraw
4. Compute Interest (Savings only)
5. Exit
Choose an option: 1
Enter amount to deposit: 999999999999
Deposited: 9.9999999999E10

Menu:
1. Deposit
2. Display Balance
3. Withdraw
4. Compute Interest (Savings only)
5. Exit
Choose an option: 2
Current balance: 9.9999999999E10

Menu:
1. Deposit
2. Display Balance
3. Withdraw
4. Compute Interest (Savings only)
5. Exit
Choose an option: 3
Enter amount to withdraw: 333333333
Withdrawn: 3.33333333E8

Menu:
1. Deposit
2. Display Balance
3. Withdraw
4. Compute Interest (Savings only)
5. Exit
Choose an option: 4
Deposited: 4.9833333333E9
Interest deposited: 4.9833333333E9

Menu:
1. Deposit
2. Display Balance
3. Withdraw
4. Compute Interest (Savings only)
5. Exit
Choose an option: 5
Exiting...
PS C:\Users\STUDENT\Desktop\LAB 5>
```

## Program : 6 Packages

Algorithm:

Create a package CIE which has two classes - student and Internals. The class student has member like usn, name, sem. The class Internal derived from student 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 in five courses of the current semester of the student. Import the two package in a file that declares the final marks of n student in all five courses.

→ Package CIE;

Public class Internal extends student {

private int [ ] internalMarks;

Public Internals (string usn, string name, int sem, int [ ] Super (usn, name, sem);

this. internalMarks = internalMarks; }

{

Public int [ ] get InternalMarks () {

return internalMarks; }

}

Public class External extends student {

Private string [ ] externalMarks;

Public External (string usn, string name, int sem) :

super (usn, name, sem), externalMarks (new int [5]) {

for (int i = 0; i < 5; i++) externalMarks [i] = 0;

System.out.println ("Enter " + i + " marks : ");

Scanner sc = new Scanner (System.in);

externalMarks [i] = sc.nextInt(); }

System.out.println ("Your marks are : ");

for (int i = 0; i < 5; i++) System.out.print (externalMarks [i] + " ");

SEE (External.java)

Package SEE;

Import CIE.Personal;

Public class External extends student{

int () external mark = new int[5];

Public External (String name, String name)

int sem, int [5] external mark; {

Super (name, name, sem);

this . external marks = external marks;

8  
Public void display External marks() {

System.out.println ("External marks : ");

for (int marks : external marks) {

System.out.print ("  
8  
" + marks + " ");

~~System.out.println ();~~

import CIE.Internalmarks;

import SEE.ExternalMarks;

Public class main {

PSVM (String [5] args) {

int n=2;

int [5] internal marks1 = {20, 30, 25, 23, 22};

int [5] external marks = {60, 70, 50, 65, 50};

External Student1 External = new External

("1BM23CS124", "Amit", 3, internal marks1,

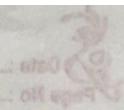
int [5] internal marks2 = {12, 25, 20, 23, 28};

int [5] external marks2 = {50, 65, 60, 52, 45};

Internal Student2 Internal = new Internal

("1BM23CS001", "Auditi", 3, internal marks2);

Scout ("Student1 Info : ");



Student Internal.display student info();  
student /Internal. display internal marks();  
Student External. display External marks();  
int [ ] final marks = calculate final Marks  
(student )internal Internal Marks; student /External  
display final marks (final mark) \* External mark);  
{

public static int () calculate final mark  
(int [ ] internal marks, int [ ] External marks)  
{ int () final marks = new int (8);  
for (int i=0; i < 8; i++) {  
final marks [i] = internal marks [i] + external  
marks [i];  
return final marks;

{  
public static void display final marks  
(int [ ] final marks) {  
cout ("final marks (internal+External): ");  
for (int mark : final marks) {  
cout (mark + " ");

~~cout~~ → System.out.println();

Output :

Enter num of std : 2

Enter name, usn, sem of std

Student name : kallian kallayat

USN : 101

Sem : 3

Enter cie marks for 5 subjects

80 75 90 85 70

Enter sec marks for 3 subjects

160 180 180 130 140

Final marks 200 100 100 100

160 150 180 150 140

Enter name usn, sem of student

B05 102 2

Student name : B05

USN : 102

Semester : 2

Enter cie marks for 5 subjects

45 80 50 45 50

Enter sec marks for 5 subjects

95 100 100 96 98

Final marks 200

92 80 100 93 97

Code:

```
import CIE.Internals;
import SEE.External;

import java.util.Scanner;

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

        Internals[] internals = new Internals[n];
        External[] externals = new External[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for Student " + (i + 1) + ":");
            System.out.print("USN: ");
            String usn = scanner.next();
            System.out.print("Name: ");
            String name = scanner.next();
            System.out.print("Semester: ");
            int sem = scanner.nextInt();

            // Input internal marks
            int[] internalMarks = new int[5];
            System.out.println("Enter internal marks for 5 courses:");
            for (int j = 0; j < 5; j++) {
                internalMarks[j] = scanner.nextInt();
            }

            // Input external marks
            int[] externalMarks = new int[5];
```

```

System.out.println("Enter external marks for 5 courses:");
for (int j = 0; j < 5; j++) {
    externalMarks[j] = scanner.nextInt();
}

// Create objects
internals[i] = new Internals(usn, name, sem, internalMarks);
externals[i] = new External(usn, name, sem, externalMarks);
}

// Display final marks
System.out.println("\nFinal Marks of Students:");
for (int i = 0; i < n; i++) {
    System.out.println("Student: " + internals[i].getName() + " (" +
internals[i].getUsn() + ")");
    System.out.println("Internal Marks: ");
    int[] internalMarks = internals[i].getInternalMarks();
    for (int mark : internalMarks) {
        System.out.print(mark + " ");
    }
    System.out.println();

    System.out.println("External Marks: ");
    int[] externalMarks = externals[i].getExternalMarks();
    for (int mark : externalMarks) {
        System.out.print(mark + " ");
    }
    System.out.println();
}

// Calculate total marks
int total = 0;
for (int mark : internalMarks) {
    total += mark;
}
for (int mark : externalMarks) {

```

```

        total += mark;
    }
    System.out.println("Total Marks: " + total);
    System.out.println();
}

scanner.close();
}
}

```

```

PS D:\project> javac CIE/*.java SEE/*.java Main.java
PS D:\project> java Main
Enter the number of students: 1
Enter details for Student 1:
USN: 78
Name: aziz
Semester: 3
Enter internal marks for 5 courses:
87
38
87
54
35
Enter external marks for 5 courses:
85
65
36
36
65

Final Marks of Students:
Student: aziz (78)
Internal Marks:
87 38 87 54 35
External Marks:
85 65 36 36 65
Total Marks: 588

PS D:\project>

```

Program :7

Exceptional Handling  
Algorithm:

Date: \_\_\_\_\_  
Page No.: \_\_\_\_\_

Week 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "father" and derived class "son" which extends base class. In son class, implement a constructor that uses both father and son's age and throws an exception if father's age <= son's age.

```

import java.util.*;
class WrongAgeException extends Exception {
    public WrongAgeException (String message) {
        super (message);
    }
}
class father {
    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 {
    int sonAge;
    public son (int fatherAge, int sonAge) throws WrongAgeException, SonAgeException {
        super (fatherAge);
        if (sonAge <= fatherAge) {
    
```

throw new SonAgeException ("son's age can't be greater than or equal to father's age")

g

this.sonAge = sonAge;

g

public int getSonAge() {  
 return sonAge;

g

public String arg[] {

while (true) {

Scanner sc = new Scanner (System.in);  
System.out.println ("Enter father age ?");

int fAge = sc.nextInt();

System.out.println ("Enter son's age ?");

int sAge = sc.nextInt();

try {

Son Son = new Son (fAge, sAge);

System.out.println ("Accepted successfully.");

g

catch (WrongAgeException e) {

System.out.println (e.getMessage());

g

{

{

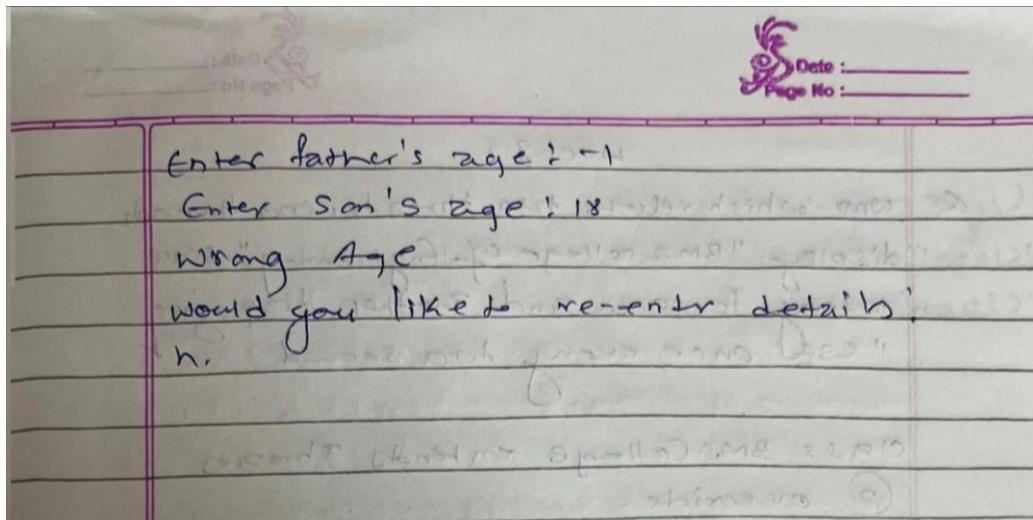
Output

Enter father age : 45

Enter son age : 18

Accepted successfully

would you like to re-enter details ? y



## Code

```
import java.util.Scanner;

class WrongAge extends Exception {

    public WrongAge() {
        super("Age error occurred");
    }

    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(); // Display father's age
    System.out.println("Son's age: " + sonAge);
}
}

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

```

```

        Son son = new Son();
        son.display();
    } catch (WrongAge e) {
        System.out.println("Exception: " + e.getMessage());
    }
}
}

```



```

44.6 KB

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Asus\Desktop\lab 7> javac ExceptionHandling.java
PS C:\Users\Asus\Desktop\lab 7> java ExceptionHandling
Enter father's age: 78
Enter son's age: 35
Father's age: 78
Son's age: 35
PS C:\Users\Asus\Desktop\lab 7> java ExceptionHandling
Enter father's age: 78
Enter son's age: -35
Exception: Age cannot be negative
PS C:\Users\Asus\Desktop\lab 7> |

```

## Program 8 Threads

Algorithm:

## Week 8

- Q WAP which releases two threads, one thread displays "BMS college of Engineering" since every ten sec and another displaying "CSE" once every two second

class BMS College extends Thread

② override

```
public void run() {
```

```
try {
```

```
while (true) {
```

```
System.out.println(" BMS college of Engineering ");  
Thread.sleep(10000);
```

```
}
```

```
} catch (InterruptedException e) {
```

```
System.out.println(e);
```

```
}
```

```
}
```

class CSE Thread extends Thread {

② override

```
public void run() {
```

```
try {
```

```
while (true) {
```

```
System.out.println(" CSE ");  
Thread.sleep(2000);
```

```
}
```

```
} catch (InterruptedException e) {
```

```
System.out.println(e);
```

```
}
```

```
}
```

Public class main {

public static void main (String [3] args)

BMScollege bms threads = new BMScollege();

CSE Thread CSE thread = new CSEThread();

bms Thread = BMSThread();

cse Thread.start();

bms Thread.start();

Output

CSE

CSE

CSC

BMS college of Engineering

CSE

ESE

CSE

BMS college of Engineering

" " " "

~~obtained "BMS college of Engineering"~~

~~obtained "BMS college of Engineering"~~

~~10  
10~~

~~2/2/24 - BMS~~

~~obtained "BMS college of Engineering"~~

Code:

```
public class Main {  
  
    static class BMSPrinter extends Thread {  
        public void run() {  
            while (true) {  
                System.out.println("BMS College of Engineering");  
                try {  
                    Thread.sleep(10000);  
                } catch (InterruptedException e) {  
                    System.out.println(e);  
                }  
            }  
        }  
    }  
  
    static class CSEPrinter extends Thread {  
        public void run() {  
            while (true) {  
                System.out.println("CSE");  
                try {  
                    Thread.sleep(2000);  
                } catch (InterruptedException e) {  
                    System.out.println(e);  
                }  
            }  
        }  
    }  
  
    public static void main(String[] args) {  
        BMSPrinter thread1 = new BMSPrinter();  
        CSEPrinter thread2 = new CSEPrinter();  
  
        thread1.start();  
        thread2.start();  
    }  
}
```

```
}
```

```
}
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Asus\Desktop\lab 8> javac Main.java
PS C:\Users\Asus\Desktop\lab 8> java Main
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
```

Program :9

Open Ended Question ( Division App)

Algorithm:

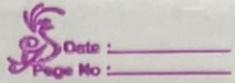
## Week 9

Q. WAP that creates a user interface to perform integer divisions. User enters two numbers num1 and num2. Division of num1 and num2 is displayed on Result field when Divide button is clicked. If num2 is zero then program would throw an arithmetic exception.

```

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

class SwingDemo {
    SwingDemo() {
        JFrame frm = new JFrame("Divide APP");
        frm.setSize(275, 150);
        frm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel lab = new JPanel("Enter divisor  
and dividend");
        JTextField zjtf = new JTextField(2);
        JTextField zf = new JTextField(8);
        JButton new = new JButton("calculate");
        JLabel enr = new JLabel();
        JLabel lab1 = new JLabel();
        JLabel blab = new JLabel();
        JLabel enslab = new JLabel();
        frm.add(enr);
        frm.add(zf);
        frm.add(zjtf);
        frm.add(new);
        frm.add(lab1);
        frm.add(blab);
        frm.add(enslab);
    }
}
    
```



```
j) from . add (blab);  
j) from . add (anolab);  
aift . add Action listem (1);  
bift . add Action (istem (1));  
button add Action listem (new Action  
listem ()) {  
    public void actionperformed (ActionEvent  
e) {  
        System.out.println ("");  
    }  
}
```

System.out.println("Action event from a text field");

ajt. add Actions (listens (1));  
bjt. add Action (listens (2));

button. addActionListeners(new ActionListener[]{})) {

public void actions\_performed (Actions(Event evt)) {

try { int a = Integer.parseInt((String)f.get(2)); }

`int b = Integer.parseInt(b);`

$\Rightarrow 1 = b : \text{sat} \text{ fact} ("1 = b = " + z);$

b ~~lax~~. setText ("In B = "+b);

An Slab set Text ("In AS = "+am);

9

```
catch ( NumberFormatException e ) {
```

```
    el.set_text(" ");
```

```
blank.setText(" ")'
```

anlab.setLabel("The name is " + name);

... setText("In 1937",

3. *What is the main idea of the story?*

) from .Set Visible (true),

5-18

```
    sum (string, args [ ]) {
```

Public void num1() {

new swing Demolition

Output

Q) Num : 10

Num : 2

Result : 5

Q) Num : 10.5

Num : 2

"Please enter valid integers"

"("block code is marked because it is not visible in the image")"

Q) Num : 10

Num : 0

"Division by zero not allowed"

10

2/0 // 20

? ("Enter first number : ")

"10" // 10

" " // 0

"0" // 0

"("sum") = 10.0" // 10.0

? ("("sum") = 10.0" // 10.0)

? ("("sum") = 10.0" // 10.0)

"("sum") = 10.0" // 10.0

Code :

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

public class IntegerDivisionApp {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Integer Division Calculator");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);

        JLabel labelNum1 = new JLabel("Num1:");
        JTextField textNum1 = new JTextField(10);

        JLabel labelNum2 = new JLabel("Num2:");
        JTextField textNum2 = new JTextField(10);

        JButton buttonDivide = new JButton("Divide");
        JLabel labelResult = new JLabel("Result: ");
        JTextField textResult = new JTextField(10);
        textResult.setEditable(false);

        JPanel panel = new JPanel(new GridLayout(4, 2));
        panel.add(labelNum1);
        panel.add(textNum1);
        panel.add(labelNum2);
        panel.add(textNum2);
        panel.add(buttonDivide);
        panel.add(labelResult);
        panel.add(textResult);

        frame.add(panel);

        buttonDivide.addActionListener(new ActionListener() {
```

```

@Override
public void actionPerformed(ActionEvent e) {
    try {
        String num1Str = textNum1.getText();
        String num2Str = textNum2.getText();

        int num1 = Integer.parseInt(num1Str);
        int num2 = Integer.parseInt(num2Str);

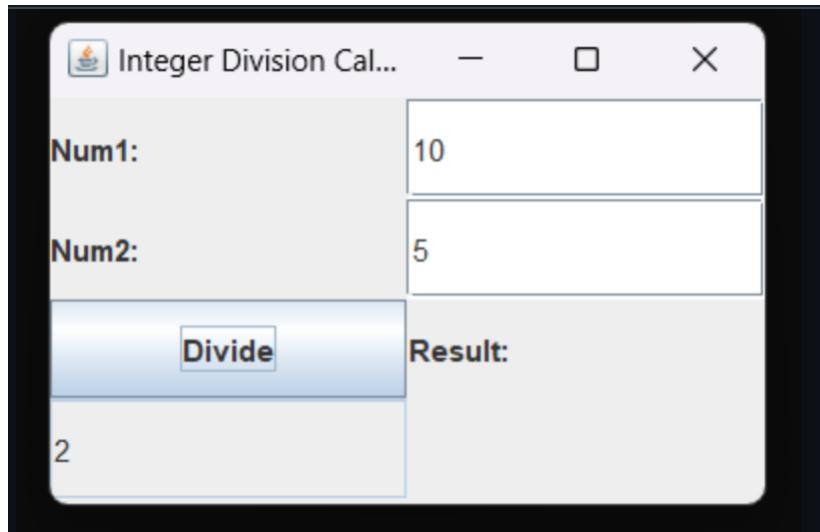
        if (num2 == 0) {
            throw new ArithmeticException("Cannot divide by zero.");
        }

        int result = num1 / num2;
        textResult.setText(String.valueOf(result));

    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(frame, "Invalid input! Please
enter valid integers.", "Input Error", JOptionPane.ERROR_MESSAGE);
    } catch (ArithmeticException ex) {
        JOptionPane.showMessageDialog(frame, ex.getMessage(),
"Arithmetic Error", JOptionPane.ERROR_MESSAGE);
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(frame, "An unexpected error
occurred: " + ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}

frame.setVisible(true);
}
}

```



Program :10

Open Ended Exercise (Inter Process Communcation And Deadlock )

Algorithm:

Page No. \_\_\_\_\_

Week 10

Q. Demonstrate Inter process Communication and dead lock

```

import java.util.*;
class A {
    synchronized void fav(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A. fav()");
        try {
            Thread.sleep(10000);
        } catch (Exception e) {
            System.out.println("A is interrupted");
        }
        System.out.println(name + " trying to call B. last()");
        b.last();
    }
}

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

class B {
    synchronized void fav(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B. fav()");
        try {
            Thread.sleep(10000);
        } catch (Exception e) {
            System.out.println("B is interrupted");
        }
    }
}

```

Date : \_\_\_\_\_  
Page No. : \_\_\_\_\_

```

System.out.println("Name: " + name + " trying to call A.123+()",
    a.123+());
}
void test() {
    System.out.println("Inside A.123+");
}

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

public static void main(String args[]) {
    new Deadlock();
}

Output: Main thread entered A-test
        Racing thread entered B-bar
        main thread trying to call B.123+()
        Inside A.123+
        Inside A.123+
        Back in other thread.

```

Open Ended Exercise.

Demonstrating Inter process Communication  
 P Deadlocks

IPC

1) Classes CSE

int n;

boolean volumeSet = false;

Synchronized int gate

while (!volumeSet)

try

System.out.println ("In Consumer waiting In");  
 wait();

{ catch (InterruptedException) {

System.out.println ("Intercepted Exception caught");

System.out.println ("not !:" + n);

volumeSet = true;

System.out.println ("In is Semaphore released");

notify();

return n;

}

Synchronized void put (int n) {

while (volumeSet)

try

System.out.println ("In Producer waiting In");

wait();

{ catch (InterruptedException) {

System.out.println ("Intercepted Exception caught");

}

This.n = n;

volumeSet = true;

```

System.out.println("put : " + n);
System.out.println("In Intimate Consumer\n");
notify();
}
}

```

Class producer implements Runnable {  
 Q9;  
 produces (Q9) {

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

Public void run() {

i = 0;

while (i < 5) {

q.put(i++);

}

}

\* Class consumer implement Runnable {

Q9;

Consumer (Q9) {

this.q = q;

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

Public void run() {

i = 0;

while (i < 5) {

int s = q.get();

System.out.println("Consumer read " + s + "\n");

i++;

}

}

```
class P(Producer){  
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

Notify consumer  
~~producer waiting~~  
get : 0

Notify producer  
put : 1

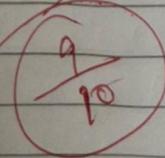
Consumed : 0

Notify consumer  
~~producer waiting~~  
get : 1

Notify producer  
put : 2

Consumed : 1

ST  
2/12/2018



Code

Dead lock

package Lab;

```
class A {  
    synchronized void foo(B b) {  
        String name = Thread.currentThread().getName();  
        System.out.println(name + " entered A.foo");  
  
        try {  
            Thread.sleep(1000); // This may throw InterruptedException  
        } catch (InterruptedException 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); // This may throw InterruptedException  
        } catch (InterruptedException 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");
    }
}

public class Deadlock implements Runnable {
    A a = new A();
    B b = new B();

    Deadlock() {
        // Start the thread and set its name
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();

        // Main thread acquires lock on a and calls foo
        a.foo(b);

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

    @Override
    public void run() {
        // This method runs in the new thread
        b.bar(a);
    }
}

public static void main(String[] args) {
    // Create the Deadlock instance and trigger the deadlock scenario
    new Deadlock();
}

```

```
}
```

```
"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.3\lib\idea_rt.jar=61106:C:\Program Files\JetBrains\In
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

Process finished with exit code 0
```

IPS

```
package Lab2;
```

```
class A {
    synchronized void foo(Lab2.B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");

        try {
            Thread.sleep(1000); // This can throw InterruptedException
        } catch (InterruptedException 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(Lab2.A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.bar");

        try {
            Thread.sleep(1000); // This can throw InterruptedException
        } catch (InterruptedException 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");
}
}

public class IPS implements Runnable {
    Lab2.A a = new Lab2.A();
    Lab2.B b = new Lab2.B();

    IPS() {
        // Set up the main thread and start the new thread
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
    }
}

```

```

// Main thread acquires lock on object a and calls foo
a.foo(b);

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

@Override
public void run() {
    // This method will be run in the RacingThread
    b.bar(a);
}

public static void main(String[] args) {
    // The main method will create an instance of IPS and trigger the
deadlock
    new IPS();
}
}

```

```

"C:\Program Files\Java\jdk-23\bin\java.exe" "-javaagent:C:\P
MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
Inside A.last
MainThread trying to call B.last()
Inside B.last
Back in main thread

Process finished with exit code 0

```