

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



LAB REPORT
on
Object Oriented Java Programming
(23CS3PCOOJ)

Submitted by

Lakshay juneja (**1BM23CS163**)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

Sep-2024 to Jan-2025

**B.M.S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering**



CERTIFICATE

This is to certify that the Lab work entitled “Object Oriented Java Programming (23CS3PCOOJ)” carried out by **Lakshay juneja (1BM23CS163)**, who is bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Ambuja K Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
---	---

Index

Sl. No.	Date	Experiment Title	Page No.
1	30/09/2024	Program 1 - Implement Quadratic Equation	04
2	7/10/2024	Program 2 - SGPA	06
3	14/10/2024	Program 3 - Books	10
4	21/10/2024	Program 4 - Shapes	15
5	28/10/2024	Program 5 - Bank	22
6	11/11/2024	Program 6 - SEE, CIE Package	28
7	28/11/2024	Program 7 - Father , Son Exception	38
8	28/11/2024	Program 8 - Threads	40
9	28/11/2024	Program 9 - Division app	43
10	28/11/2024	Program 10 - DeadLock & IPC	46

Github Link:

<https://github.com/Lakshay1518/JAVA-LAB-PROGRAM->

Program 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

Algorithm:

LAB-1 Date 30/10/24 Page 1/1

Program 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
```

```
public class QuadraticEquationSolver {
```

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

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

```
        System.out.print("Enter coefficient a: ");
```

```
        double a = scanner.nextDouble();
```

```
        if (a == 0) {
```

```
            System.out.println("Not a quadratic equation");
```

```
            System.out.print("Enter a non-zero value for a: ");
```

```
            a = scanner.nextDouble();
```

```
        }
```

```
        System.out.print("Enter coefficient b: ");
```

```
        double b = scanner.nextDouble();
```

```
        System.out.print("Enter coefficient c: ");
```

```
        double c = scanner.nextDouble();
```

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

```
        if (d == 0) {
```

```
            double root = -b / (c * a);
```

```
            System.out.println("Roots are real and equal");
```

```
            System.out.println("Root 1 and Root 2: " + root);
```

```
        } else if (d > 0) {
```

```
            double r1 = (-b + Math.sqrt(d)) / (2 * a);
```

```
            double r2 = (-b - Math.sqrt(d)) / (2 * a);
```

```
            System.out.println("Roots are real and distinct");
```

```
            System.out.println("Root 1: " + r1);
```

```
            System.out.println("Root 2: " + r2);
```

```
        } else {
```

```
            double realPart = -b / (2 * a);
```

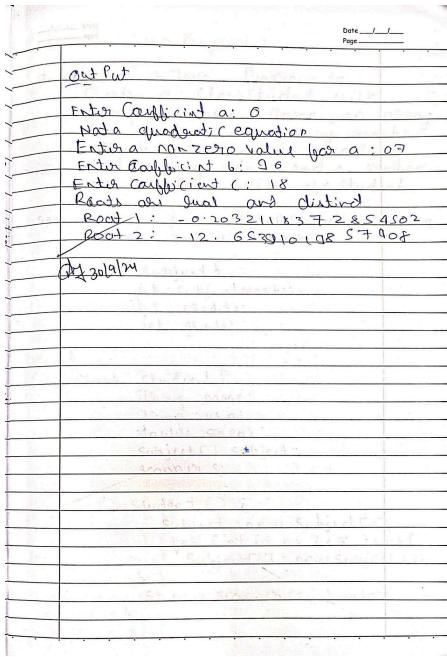
```
            double imaginaryPart = Math.sqrt(-d) / (2 * a);
```

```
            System.out.println("Roots are imaginary");
```

```
            System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
```

```
            System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");
```

```
        }
```



Code:

```
import java.util.Scanner;

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

        System.out.print("Enter coefficient a: ");
        double a = scanner.nextDouble();

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

        System.out.print("Enter coefficient b: ");
        double b = scanner.nextDouble();
        System.out.print("Enter coefficient c: ");
        double c = scanner.nextDouble();

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

        if (d == 0) {
            double root = -b / (2 * a);
            System.out.println("Roots are real and equal");
        }
    }
}
```

```

        System.out.println("Root 1 and Root 2: " + root);
    } else if (d > 0) {
        double r1 = (-b + Math.sqrt(d)) / (2 * a);
        double r2 = (-b - Math.sqrt(d)) / (2 * a);
        System.out.println("Roots are real and distinct");
        System.out.println("Root 1: " + r1);
        System.out.println("Root 2: " + r2);
    } else {
        double realPart = -b / (2 * a);
        double imaginaryPart = Math.sqrt(-d) / (2 * a);
        System.out.println("Roots are imaginary");
        System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
        System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");
    }
}
}
}
}

```

```

C:\Windows\System32\cmd.exe x + v
(c) Microsoft Corporation. All rights reserved.

D:\1BM23CS163>javac QuadraticEquationSolver.java

D:\1BM23CS163>java QuadraticEquationSolver
Enter coefficient a: 0
Not a quadratic equation
Enter a non-zero value for a: 5
Enter coefficient b: 6
Enter coefficient c: 9
Roots are imaginary
Root 1: -0.6 + 1.2i
Root 2: -0.6 - 1.2i

D:\1BM23CS163> javac QuadraticEquationSolver.java
D:\1BM23CS163>^X

D:\1BM23CS163>java QuadraticEquationSolver
Enter coefficient a: 0
Not a quadratic equation
Enter a non-zero value for a: 18
Enter coefficient b: 07
Enter coefficient c: 45
Roots are imaginary
Root 1: -0.1944444444444445 + 1.5691371380554i
Root 2: -0.1944444444444445 - 1.5691371380554i

D:\1BM23CS163>java QuadraticEquationSolver
Enter coefficient a: -08
Enter coefficient b: 73
Enter coefficient c: 98
Roots are real and distinct
Root 1: -1.1878396638807347
Root 2: 10.312839663880734

D:\1BM23CS163>

D:\1BM23CS163>java QuadraticEquationSolver
Enter coefficient a: 30
Enter coefficient b: 81
Enter coefficient c: 69
Roots are imaginary
Root 1: -1.35 + 0.6910137480542626i
Root 2: -1.35 - 0.6910137480542626i

D:\1BM23CS163> NAME = LAKSHAY JUNEJA

```

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:-

Lab Program - 2

```

1. 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 CGPA of a student

2. import java.util.InputMismatchException;
   import java.util.Scanner;

3. class Subject {
        int subjectMarks;
        int credits;
        int grade;
    }

4. class Student {
        String name;
        String USN;
        double CGPA;
        Subject[] subject;
        Scanner s;
    }

5. Student() {
    subject = new Subject[5];
    for (int i = 0; i < 5; i++) {
        Subject[i] = new Subject();
    }
}

6. public static void main(String args[]) {
    Scanner s = new Scanner(System.in);
    Student s1 = new Student();
    System.out.print("Enter USN: ");
    USN = s.nextLine();
    System.out.print("Enter Name: ");
    name = s.nextLine();

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

    void getMarks() {
        for (int i = 0; i < 5; i++) {
            while (true) {
                try {
                    System.out.print("Enter marks for Subject " + (i + 1) + ": ");
                    Subject[i].marks = s.nextInt();
                    if (Subject[i].marks >= 0) {
                        break;
                    } else {
                        System.out.println("Please enter a non-negative number.");
                    }
                } catch (InputMismatchException e) {
                    System.out.println("Invalid input. Please enter a number.");
                    s.next();
                }
            }
        }
    }

    void calculateCGPA() {
        double totalCredits = 0;
        double totalGradePoints = 0;
        for (int i = 0; i < 5; i++) {
            totalCredits += Subject[i].credits;
            totalGradePoints += Subject[i].grade * Subject[i].credits;
        }
        CGPA = totalGradePoints / totalCredits;
    }

    public static void main(String args[]) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.calculateCGPA();
        System.out.println("Name: " + s1.name);
        System.out.println("USN: " + s1.USN);
        System.out.println("CGPA: " + s1.CGPA);
    }
}

```

3 else {
 break;
}
3 catch (InputMismatchException e){
 System.out.println("Invalid input. Please enter a number.");
 s.next();
}
if (Subject[i].subjectMarks < 10) {
 Subject[i].grade = 0;
} else if (Subject[i].subjectMarks >= 10) {
 Subject[i].grade = 10;
} else if (Subject[i].subjectMarks >= 80) {
 Subject[i].grade = 9;
} else if (Subject[i].subjectMarks >= 70) {
 Subject[i].grade = 8;
} else if (Subject[i].subjectMarks >= 60) {
 Subject[i].grade = 7;
} else if (Subject[i].subjectMarks >= 50) {
 Subject[i].grade = 6;
} else {
 Subject[i].grade = 5;
}
4
4
4

void calculateCGPA() {
 double totalCredits = 0;
 double totalGradePoints = 0;
 for (int i = 0; i < 5; i++) {
 totalCredits += Subject[i].credits;
 totalGradePoints += Subject[i].grade * Subject[i].credits;
 }
 CGPA = totalGradePoints / totalCredits;
}

public class Main {
 public static void main(String args[]) {
 Student s1 = new Student();
 s1.getStudentDetails();
 s1.getMarks();
 s1.calculateCGPA();
 System.out.println("Name: " + s1.name);
 System.out.println("USN: " + s1.USN);
 System.out.println("CGPA: " + s1.CGPA);
 }
}

Add file to project

Code:

```
import java.util.InputMismatchException;
import java.util.Scanner;
```

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

```

}

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

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

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

    void getMarks() {
        for (int i = 0; i < 9; i++) {
            while (true) {
                try {
                    System.out.println("Enter marks for subject " + (i + 1) + ": ");
                    subject[i].subjectMarks = s.nextInt();
                    if (subject[i].subjectMarks > 100 || subject[i].subjectMarks < 0) {
                        System.out.println("Invalid marks. Please enter marks between 0 and 100.");
                    } else {
                        break;
                    }
                } catch (InputMismatchException e) {
                    System.out.println("Invalid input. Please enter a number.");
                    s.next(); // Clear the invalid input
                }
            }
        }
    }

    while (true) {
        try {
            System.out.println("Enter credits for subject " + (i + 1) + ": ");
            subject[i].credits = s.nextInt();
            if (subject[i].credits < 0) {
                System.out.println("Invalid credits. Please enter a non-negative number.");
            } else {

```

```

        break;
    }
} catch (InputMismatchException e) {
    System.out.println("Invalid input. Please enter a number.");
    s.next(); // Clear the invalid input
}
if (subject[i].subjectMarks < 40) {
    subject[i].grade = 0;
} else if (subject[i].subjectMarks >= 90) {
    subject[i].grade = 10;
} else if (subject[i].subjectMarks >= 80) {
    subject[i].grade = 9;
} else if (subject[i].subjectMarks >= 70) {
    subject[i].grade = 8;
} else if (subject[i].subjectMarks >= 60) {
    subject[i].grade = 7;
} else if (subject[i].subjectMarks >= 50) {
    subject[i].grade = 6;
} else {
    subject[i].grade = 5;
}
}

void computeSGPA() {
    double totalCredits = 0;
    double totalGradePoints = 0;
    for (int i = 0; i < 9; i++) {
        totalCredits += subject[i].credits;
        totalGradePoints += subject[i].grade * subject[i].credits;
    }
    SGPA = totalGradePoints / totalCredits;
}

public class Main {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        System.out.println("Name: " + s1.name);
        System.out.println("USN: " + s1.usn);
        System.out.println("SGPA: " + s1.SGPA);
    }
}

```

```
C:\Windows\System32\cmd.exe + ^

Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

D:\LAB 2 SGPA>javac Main.java

D:\LAB 2 SGPA>java Main
Enter USN:
1bm23cs163
Enter Name:
lakshay juneja
Enter marks for subject 1:
Invalid input. Please enter a number.
Enter marks for subject 1:
89
Enter credits for subject 1:
4
Enter marks for subject 2:
34
Enter credits for subject 2:
1
Enter marks for subject 3:
67
Enter credits for subject 3:
1
Enter marks for subject 4:
67
Enter credits for subject 4:
2
Enter marks for subject 5:
69
Enter credits for subject 5:
4
Enter marks for subject 6:
90
Enter credits for subject 6:
4
Enter marks for subject 7:
23
Enter credits for subject 7:
1
Enter marks for subject 8:
56
Enter credits for subject 8:
4
Enter marks for subject 9:
90
Enter credits for subject 9:
3
Name: lakshay
USN: 1bm23cs163
SGPA: 7.45833333333333
```

Program - 3

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the

values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

Algorithm:-

Date / /
Page / /

Q. Develop a Java program to create a class `Student` with member `usn`.
class `Book` which contains four
members: `name`, `author`, `price`,
`numPages`. Include a constructor
to set the values for members.
Include methods to set and get
the details of the objects. Include
a `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 Books {
    String name;
    String author;
    int price;
    int numPages;
    Books (String name, String author,
           int price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public void +toString () {
        System.out.println("The name of
                           the book : " + this.name);
        System.out.println("The author of book :
                           " + this.author);
    }
}
```

	Date _____	Page _____

```

    system.out.println("The price of the book: " + this.price);
    system.out.println("The number of pages: " + this.numPages);
}

class Student {
    public static void main (String args) {
        int n = 6;
        system.out.println("Enter the numbers of books:");
        Scanner sc = new Scanner (System.in);
        n = sc.nextInt();
        Books [ ] b = new Books[n];
        sc.nextLine();
        for (int i = 0; i < n; i++) {
            system.out.println("Enter the name, author, price, and number of pages for book" + (i+1));
            String name = sc.nextLine();
            String author = sc.nextLine();
            int price = sc.nextInt();
            int numPages = sc.nextInt();
            sc.nextLine();
            b[i] = new Book(name, author, price, numPages);
        }
        system.out.println("Book details:");
        for (int i = 0; i < n; i++) {
            System.out.println(b[i].toString());
        }
    }
}

```

import java.util.Scanner;

```

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

```

```

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

```

```
@Override  
public String toString() {  
    return "Book name: " + name +  
    "\nAuthor: " + author + "\nPrice: " + price  
    + "\nPages: " + numPages + "\n";  
}  
}
```

```
class Main { // Removed the public access  
modifier  
    public static void main(String[] args) {  
        Scanner scanner = new  
Scanner(System.in);  
        System.out.print("Enter number of  
books: ");  
        int n = scanner.nextInt();  
        scanner.nextLine(); // Consume  
newline
```

```
        Book[] books = new Book[n];  
        for (int i = 0; i < n; i++) {  
            System.out.println("Enter details  
for book " + (i + 1) + ":");  
            System.out.print("Name: ");  
            String name = scanner.nextLine();  
            System.out.print("Author: ");  
            String author = scanner.nextLine();  
            System.out.print("Price: ");  
            int price = scanner.nextInt();
```

```
        System.out.print("Number of  
        pages: ");  
        int numPages = scanner.nextInt();  
        scanner.nextLine() // Consume  
newline  
        books[i] = new Book(name,  
author, price, numPages);  
    }  
  
    System.out.println("\nBook  
Details:");  
    for (Book book : books) {  
        System.out.println(book);  
    }  
    scanner.close();  
}
```

```
C:\> C:\Windows\System32\cmd.e + ~
Enter number of books: 4
Enter details for book 1:
Name: harry potter
Author: jk
Price: 543
Number of pages: 345
Enter details for book 2:
Name: think and grow rich
Author: lakshay
Price: 87
Number of pages: 90
Enter details for book 3:
Name: game of thrones
Author: avengers
Price: 654
Number of pages: 900
Enter details for book 4:
Name: how to win friends
Author: daile
Price: 546
Number of pages: 2324

Book Details:
Book name: harry potter
Author: jk
Price: 543
Pages: 345

Book name: think and grow rich
Author: lakshay
Price: 87
Pages: 90

Book name: game of thrones
Author: avengers
Price: 654
Pages: 900

Book name: how to win friends
Author: daile
Price: 546
Pages: 2324

D:\Lab 3>
```

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:-

LAB Program-4

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

```
see:- import java.util.Scanner;
abstract class Shape {
    protected int dimension1;
    protected int dimension2;
    public Shape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }
    public abstract void printArea();
}
class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        super(length, width);
    }
    @Override
    public void printArea() {
        int area = dimension1 * dimension2;
    }
}
```

Date _____
Page _____

```

public class Shape {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the length of the rectangle: ");
        int length = scanner.nextInt();
        System.out.print("Enter the width of the rectangle: ");
        int width = scanner.nextInt();
        Shape rectangle = new Rectangle(length, width);
        rectangle.printArea();
    }
}

System.out.print("Enter the base of the triangle: ");
int base = scanner.nextInt();
System.out.print("Enter the height of the triangle: ");
int height = scanner.nextInt();
Shape triangle = new Triangle(base, height);
triangle.printArea();

System.out.print("Enter the radius of circle: ");
int radius = scanner.nextInt();
Shape circle = new Circle(radius);
circle.printArea();
scanner.close();
}

```

Date _____
Page _____

```

class Rectangle extends Shape {
    public Rectangle(int base, int height) {
        super(base, height);
    }
    @Override
    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of triangle: " + area);
    }
}

class Circle extends Shape {
    public Circle(int radius) {
        super(radius);
    }
    @Override
    public void printArea() {
        double area = Math.PI * dimension * dimension;
        System.out.println("Area of circle: " + area);
    }
}

```

Date _____
Page _____

	<p><u>Output</u></p> <pre> class Shape { protected int dimension1; protected int dimension2; } public Shape(int dimension1, int dimension2) { this.dimension1 = dimension1; this.dimension2 = dimension2; } class Rectangle extends Shape { public void calculateArea() { int length = dimension1; int width = dimension2; int area = length * width; System.out.println("Area of Rectangle: " + area); } } class Triangle extends Shape { public void calculateArea() { int base = dimension1; int height = dimension2; double area = (base * height) / 2; System.out.println("Area of Triangle: " + area); } } class Circle extends Shape { public void calculateArea() { double radius = dimension1; double pi = 3.14; double area = pi * radius * radius; System.out.println("Area of Circle: " + area); } } public class Main { public static void main(String[] args) { Shape rectangle = new Rectangle(7, 4); rectangle.calculateArea(); Shape triangle = new Triangle(5, 4); triangle.calculateArea(); Shape circle = new Circle(6); circle.calculateArea(); } } </pre> <p>Enter length of rectangle: 7 Enter width of rectangle: 4 Area of rectangle: 28 Enter the base of triangle 5 Enter the height of triangle 4 Area of triangle: 10 Enter radius of circle 6 Area of circle: 113.047335</p> <p><u>Conclusion</u></p> <p>Program output is correct.</p> <p>Ques. What is the output of the following program?</p> <pre> class Shape { protected int dimension1; protected int dimension2; } class Rectangle extends Shape { public void calculateArea() { int length = dimension1; int width = dimension2; int area = length * width; System.out.println("Area of Rectangle: " + area); } } class Triangle extends Shape { public void calculateArea() { int base = dimension1; int height = dimension2; double area = (base * height) / 2; System.out.println("Area of Triangle: " + area); } } class Circle extends Shape { public void calculateArea() { double radius = dimension1; double pi = 3.14; double area = pi * radius * radius; System.out.println("Area of Circle: " + area); } } public class Main { public static void main(String[] args) { Shape rectangle = new Rectangle(7, 4); rectangle.calculateArea(); Shape triangle = new Triangle(5, 4); triangle.calculateArea(); Shape circle = new Circle(6); circle.calculateArea(); } } </pre> <p>Ans. The output will be: - Area of rectangle: 28 - Area of triangle: 10 - Area of circle: 113.047335</p>
--	--

code ;

```
import java.util.Scanner;
```

```
abstract class Shape {  

    protected int dimension1;  

    protected int dimension2;
```

```
public Shape(int dimension1, int dimension2) {  

    this.dimension1 = dimension1;  

    this.dimension2 = dimension2;  

}
```

```
    public abstract void printArea();  
}  
  
}
```

```
class Rectangle extends Shape {  
    public Rectangle(int length, int width) {  
        super(length, width);  
    }  
  
    @Override  
    public void printArea() {  
        int area = dimension1 * dimension2;  
        System.out.println("Area of Rectangle: " + area);  
    }  
}
```

```
class Triangle extends Shape {  
    public Triangle(int base, int height) {  
        super(base, height);  
    }  
  
    @Override  
    public void printArea() {  
        double area = 0.5 * dimension1 * dimension2;  
        System.out.println("Area of Triangle: " + area);  
    }  
}
```

```
class Circle extends Shape {  
    public Circle(int radius) {  
        super(radius, 0); // Only radius is used, so dimension2 is  
        set to 0  
    }  
}
```

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

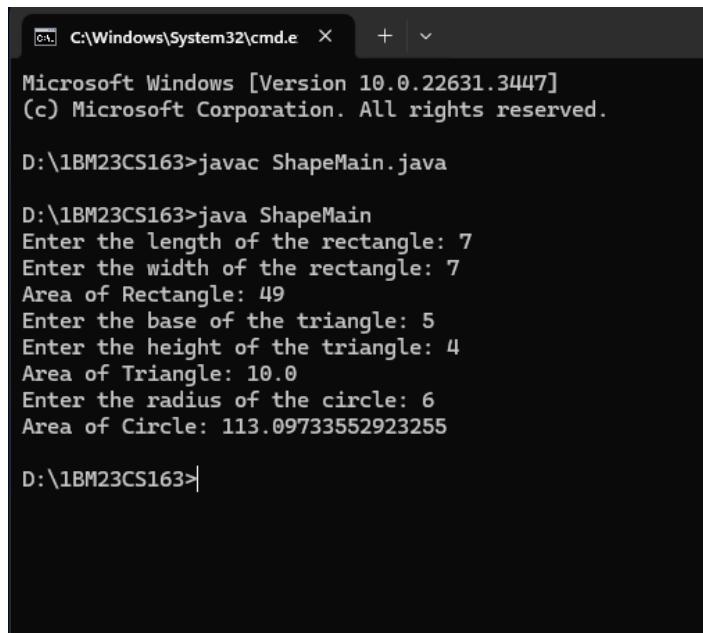
```
public class ShapeMain {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.print("Enter the length of the rectangle: ");  
        int length = scanner.nextInt();  
        System.out.print("Enter the width of the rectangle: ");  
        int width = scanner.nextInt();  
        Shape rectangle = new Rectangle(length, width);  
        rectangle.printArea();  
    }  
}
```

```
System.out.print("Enter the base of the triangle: ");  
int base = scanner.nextInt();  
System.out.print("Enter the height of the triangle: ");
```

```
int height = scanner.nextInt();
Shape triangle = new Triangle(base, height);
triangle.printArea();
```

```
System.out.print("Enter the radius of the circle: ");
int radius = scanner.nextInt();
Shape circle = new Circle(radius);
circle.printArea();
```

```
scanner.close();
}
}
```



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window displays the following output:

```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

D:\1BM23CS163>javac ShapeMain.java

D:\1BM23CS163>java ShapeMain
Enter the length of the rectangle: 7
Enter the width of the rectangle: 7
Area of Rectangle: 49
Enter the base of the triangle: 5
Enter the height of the triangle: 4
Area of Triangle: 10.0
Enter the radius of the circle: 6
Area of Circle: 113.09733552923255

D:\1BM23CS163>
```

Program 5

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

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

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

Algorithm:-

LAB Programs	
Date _____	Page _____
e) check for the minimum balance imposed if necessary at what the balance.	
import java.util.Scanner;	
abstract class Account {	
protected String customerName;	
protected String accountNumber;	
protected double balance;	
public Account (String customerName, etc;	
this.customerName = customerName;	
this.accountNumber = accountNumber;	
this.balance = 0.0;	
}	
public void deposit (double amount) {	
balance += amount;	
System.out.println ("Deposited: " + amou	
}	
public void displayBalance () {	
System.out.println ("Current balance: " +	
balance);	
}	
public abstract void withdraw (doub	
tamount);	
}	

Date _____	
Page _____	
class SavAcct extends Account {	
private double interestRate;	
public SavAcct (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);	
}	
@Override	
public void withdraw (double amount) {	
if (amount <= balance) {	
balance -= amount;	
System.out.println ("Withdrawn: " +	
amount);	
} else {	
System.out.println ("Insufficient	
balance");	
}	
}	
}	
class CurrAcct extends Account {	
private static final double MTN_BALANCE	
= 1000.0;	
private static final double SERVICE_CHARGE	
= 30.0;	
public CurrAcct (String customerName,	
String accountNumber) {	
super (customerName, accountNumber);	
@Override	
public void withdraw (double amount) {	
if (amount <= balance) {	
balance -= amount;	
System.out.println ("Withdrew: " +	
amount);	
checkMinimumBalance ();	
} else {	
System.out.println ("Insufficient	
balance");	
}	
}	
private void checkMinimumBalance () {	
if (balance < MTN_BALANCE) {	
balance -= SERVICE_CHARGE;	
System.out.println ("Service charge	
applied: " + SERVICE_CHARGE);	
}	
}	
2 hours later, amount = 1000.00	
when deposit long time, balance	

```

Date: / / Page: /
public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter customer name");
        String name = scanner.nextLine();
        System.out.println("Choose account number");
        String accNumber = scanner.nextLine();
        System.out.println("Choose account type");
        int accountType = scanner.nextInt();
        Account account;
        if (accountType == 1) {
            System.out.println("Enter interest rate");
            double interestRate = scanner.nextDouble();
            account = new SavAcc(name, accNumber, interestRate);
        } else {
            account = new CurAcc(name, accNumber);
        }
        while (true) {
            System.out.println("1. Deposit\n2. Withdraw\n3. Display Balance\n4. Exit");
            int choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    System.out.println("Enter deposit amount");
                    account.deposit(scanner.nextDouble());
                    break;
                case 2:
                    System.out.println("Enter withdraw amount");
                    account.withdraw(scanner.nextDouble());
                    break;
                case 3:
                    account.displayBalance();
                    break;
                case 4:
                    System.out.println("Exiting...");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid choice");
            }
        }
    }
}

```

```

Date: / / Page: /
output:
Enter customer name:
Jahiday Junaib
Enter account number:
1600135163
Choose account type (1 for Savings, 2 for Current):
1
Enter interest rate:
5
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
1
Choose an option:
Enter deposit amount: 2000
Deposited 2000.0
1
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
1
Choose an option:
Enter withdraw amount: 500
Withdraw 500.0
1
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
1

```

Code:

```

import java.util.Scanner;

abstract 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) {
    balance += amount;
    System.out.println("Deposited: " + amount);
}

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

public abstract void withdraw(double amount);
}

class SavAcct extends Account {
    private double interestRate;

    public SavAcct(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);
    }
}

@Override
public void withdraw(double amount) {
    if (amount <= balance) {
        balance -= amount;
    }
}

```

```

        System.out.println("Withdrawn: " + amount);
    } else {
        System.out.println("Insufficient balance.");
    }
}
}

class CurAcct extends Account {
private static final double MIN_BALANCE = 1000.0;
private static final double SERVICE_CHARGE = 50.0;

public CurAcct(String customerName, String accountNumber) {
    super(customerName, accountNumber);
}

@Override
public void withdraw(double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println("Withdrawn: " + amount);
        checkMinimumBalance();
    } else {
        System.out.println("Insufficient balance.");
    }
}

private void checkMinimumBalance() {
    if (balance < MIN_BALANCE) {
        balance -= SERVICE_CHARGE;
        System.out.println("Service charge applied: " + SERVICE_CHARGE);
    }
}

public class Bank {
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter customer name:");
}
}

```

```

String name = scanner.nextLine();
System.out.println("Enter account number:");
String accNumber = scanner.nextLine();

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

if (accountType == 1) {
    System.out.println("Enter interest rate:");
    double interestRate = scanner.nextDouble();
    account = new SavAcct(name, accNumber, interestRate);
} else {
    account = new CurAcct(name, accNumber);
}

while (true) {
    System.out.println("\n1. Deposit\n2. Withdraw\n3. Display Balance\n4.
Exit");
    System.out.print("Choose an option: ");
    int choice = scanner.nextInt();

    switch (choice) {
        case 1:
            System.out.print("Enter deposit amount: ");
            account.deposit(scanner.nextDouble());
            break;
        case 2:
            System.out.print("Enter withdrawal amount: ");
            account.withdraw(scanner.nextDouble());
            break;
        case 3:
            account.displayBalance();
            break;
        case 4:
            System.out.println("Exiting...");
            scanner.close();
            return;
    }
}

```

default:

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

```
C:\Users\STUDENT\Desktop>java Bank
Enter customer name:
lakshay juneja
Enter account number:
1bm23cs163
Choose account type (1 for Savings, 2 for Current):
2

1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Choose an option: 1
Enter deposit amount: 2000
Deposited: 2000.0

1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Choose an option: 2
Enter withdrawal amount: 1500
Withdrawn: 1500.0
Service charge applied: 50.0

1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Choose an option: 3
Current balance: 450.0

1. Deposit
2. Withdraw
3. Display Balance
4. Exit
Choose an option: 4
Exiting...
C:\Users\STUDENT\Desktop>
```

Program - 6

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

→ Lab Program 6
 Create a package CIE which has three classes Student and Internals. The class Student has members like USN, name, sem. The class Internals contains array student mark which stores internal marks scored by each student in all five courses.

Package CIE

```

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

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

    public String getName () {
        return Name;
    }

    public String getUSN () {
        return usn;
    }
}
  
```

Classmate

```

public int getSem () {
    return sem;
}
  
```

Package CIE

```

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

    public Internals (String USN, String Name, int Sem) {
        super (usn, name, sem);
        this.internalMarks = new int [5];
    }

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

Package SEE

```

import CIE.Student;

public class External extends Student {
    private int [] externalMarks;

    public External (String USN, String Name, int Sem) {
        super (usn, name, sem);
        this.externalMarks = new int [5];
    }

    public int [] getExternalMarks () {
        return externalMarks;
    }
}
  
```

**Public int CIE.getInternalMarks () {
 return internalMarks; }
 3**

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

**Public class main {
 public static void main (String [] args) {
 Scanner scanner = new Scanner (System.in);
 System.out.print ("Enter number of students : ");
 int n = scanner.nextInt ();
 Internals [] Internals = new Internals [n];
 External [] External = new External [n];
 for (int i = 0; i < n; i++) {
 System.out.println ("Enter details for student " +
 (i + 1) + " : ");
 String usn = scanner.nextLine ();
 String name = scanner.nextLine ();
 System.out.print ("USN : ");
 String USN = scanner.nextLine ();
 System.out.print ("Name : ");
 String name = scanner.nextLine ();
 System.out.print ("Semester : ");
 int sem = scanner.nextInt ();
 Internals [i] = new Internals (USN, name, sem);
 External [i] = new External (USN, name, sem);
 System.out.println ("Enter internal marks for
 5 courses : ");
 for (int j = 0; j < 5; j++) {
 System.out.print ("Subject " + (j + 1) + " : ");
 int marks = scanner.nextInt ();
 Internals [i].internalMarks [j] = marks;
 External [i].externalMarks [j] = marks;
 }
 }
 System.out.println ("Total Marks : " + total);
 System.out.println ();
 Scanner.close ();
 }

Output :

Enter Number of Students : 2
 Name : Kratika
 USN : ABC123
 Semester : 3
 Enter details for student 1 Enter CIE marks for 5 subjects
 Subject 1 : 29
 Subject 2 : 35
 Subject 3 : 36
 Subject 4 : 31
 Subject 5 : 31
 Enter details for student 2 Enter SEE marks for 5 subjects
 Subject 1 : 81
 Subject 2 : 73
 Subject 3 : 45
 Subject 4 : 77
 Subject 5 : 83
 (similar for student 2)**

classmate
Date _____
Page _____

```

@for (int i = 0; i < 5; i++) {
    internalMarks[i] = scanner.nextInt();
}

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

intake[i] = new Internals(scanner.nextInt(), scanner.nextInt(), scanner.nextInt());
externale[i] = new External(scanner.nextInt(), scanner.nextInt(), scanner.nextInt());

```

System.out.println("Final marks of students:");
 for (int i = 0; i < n; i++) {
 System.out.println("Student " + intake[i].getname() + " " + externale[i].getExternalMark());
 }
 int total = 0;
 for (int mark : internalMarks) {
 total += mark;
 }
 for (int mark : externalMarks) {
 total += mark;
 }
}

classmate
Date _____
Page _____

Total

Subject 1: 179 marks with component A failed
 Subject 2: 189 marks with component A failing
 Subject 3: 135 marks with component B failing
 Subject 4: 178 marks with component A failing
 Subject 5: 167 marks with component A failing
 Total marks with component A failing: 825 marks
 Total marks with component B failing: 450 marks
 Total failing marks: 1275 marks

Failed Students

Failed Students: 20
 Failed Students: 10
 Failed Students: 5

Failed Students

Failed Students: 20
 Failed Students: 10
 Failed Students: 5

please scroll below

// CIE

package CIE;

```

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

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

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

```

```

package CIE;

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

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

    public String getUsn() {
        return usn;
    }

    public String getName() {
        return name;
    }

    public int getSem() {

```

```

        return sem;
    }
}

// SEE

package SEE;

import CIE.Student;

public class External extends Student {
    private int[] externalMarks;

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

    public int[] getExternalMarks() {
        return externalMarks;
    }
}

// Main

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();
}
}

```

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.20314.0660]
(c) Microsoft Corporation. All rights reserved.
C:\Users\STUDENT\Desktop\18R23CS117\LAB6>javac main.java
C:\Users\STUDENT\Desktop\18R23CS117\LAB6>java main
Enter the number of students:
2
Enter details for Student1 here:
Enter your usn here:
111
Enter your name here:
Idrees
Enter your semester here:
3
Enter your CIE marks for the 5 subjects here:
Subject1:98
Subject2:99
Subject3:99
Subject4:98
Subject5:92
CIE marks are as follows:
Subject1:88
Subject2:88
Subject3:98
Subject4:95
Subject5:92
Enter the 5 SEE Marks here:
Subject1:88
Subject2:89
Subject3:98
Subject4:94
Subject5:90
Enter details for Student2 here:
Enter your usn here:
111
Enter your name here:
Harish
Enter your semester here:
3
Enter your CIE marks for the 5 subjects here:
Subject1:99
Subject2:99
Subject3:95
Subject4:91
Subject5:87
CIE marks are as follows:
Subject1:89
Subject2:99
Subject3:98
Subject4:91
Subject5:87
Enter the 5 SEE Marks here:
Subject1:99
Subject2:98
Subject3:95
Subject4:96
Subject5:92
Finalmarks of students:
Student1:
Name:Idrees
USN:117
Semester:3
The final marks of the 5 subjects are:
Subject1:88
Subject2:99
Subject3:98
Subject4:94
Subject5:91
Student2:
Name:Harish
USN:116
Semester:3
The final marks of the 5 subjects are:
Subject1:94
Subject2:98
Subject3:95
Subject4:92
Subject5:59
C:\Users\STUDENT\Desktop\18R23CS117\LAB6>
```

Program - 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called

“Father” and derived class called “Son” which extends the base class. In Father class, implement a

constructor which takes the age and throws the exception WrongAge() when the input age<0. In Son class,

implement a constructor that cases both father and son’s age and throws an exception if son’s age is
>=father’s age.

Algorithm:-

LAB Program 7	Date 1/1 Page 1/1
<p>Q. Write a Program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age<0. In Son class, implement java.util.Scanner;</p> <pre>class WrongAge extends Exception { public WrongAge() { super("Age Erroneous Age"); } } public 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(); } } class Son extends Father { public Son() throws WrongAge { Scanner s = new Scanner(System.in); System.out.print("Enter Son's Age: "); int sonAge = s.nextInt(); if (sonAge >= fatherAge) { throw new WrongAge("Age cannot be negative"); } } }</pre>	<p>Handling (WrongAge<0) { throw new WrongAge("Age cannot be negative"); } else if (sonAge == fatherAge) { throw new WrongAge("Son's age cannot be equal to father's age"); } else if (sonAge < 0) { throw new WrongAge("Age cannot be negative"); } else { System.out.println("Son's Age: " + sonAge); }</p> <pre>public void display() { System.out.println("Son's Age: " + sonAge); }</pre>

Date _____
 Page _____

```

public class ExceptionInheritanceDemo {
    public static void main(String[] args) {
        try {
            Son son = new Son(10);
            son.display();
        } catch (WrongAgeException e) {
            System.out.println("Exception Caught: " + e.getMessage());
        }
    }
}

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

class Son extends Father {
    int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be less than 0");
        }
        if (sonAge >= fatherAge) {
            throw new IllegalArgumentException("Son's age cannot be greater than or equal to Father's age");
        }
        this.sonAge = sonAge;
    }
}

```

Output:

```

Exception Caught: Son's age cannot be greater than or equal to Father's age

```

Code:

```

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("Father's age cannot be less than 0");
        }
        this.age = age;
    }
}

class Son extends Father {
    int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be less than 0");
        }
        if (sonAge >= fatherAge) {
            throw new IllegalArgumentException("Son's age cannot be greater than or equal to Father's age");
        }
        this.sonAge = sonAge;
    }
}

```

```

}
public class Main {
    public static void main(String[] args) {
        try {
            Father father = new Father(40);
            Son son = new Son(40, 20);
            System.out.println("Father's age: " + father.age);
            System.out.println("Son's age: " + son.sonAge);
        } catch (WrongAgeException e) {
            System.err.println("Exception: " + e.getMessage());
        } catch (IllegalArgumentException e) {
            System.err.println("Exception: " + e.getMessage());
        }
    }
}

```

OutPut:

```

Enter father's age: 41
Enter son's age: 60
Error: Son's age cannot be greater than or equal to father's age
Name: lakshay
USN: 1bm23cs163

```

Program 8:-

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

Algorithm:-

Lab Program no 8

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

```

1 class CollegeThread extends Thread {
2     public void run() {
3         try {
4             while (true) {
5                 System.out.println("BMS College of Engineering");
6                 Thread.sleep(10000);
7             }
8         } catch (InterruptedException e) {
9             System.out.println("CollegeThread interrupted");
10        }
11    }
12 }
13
14 class CSEThread extends Thread {
15     public void run() {
16         try {
17             while (true) {
18                 System.out.println("CSE");
19                 Thread.sleep(2000);
20             }
21         } catch (InterruptedException e) {
22             System.out.println("CSE Thread interrupted");
23        }
24    }
25 }

```

Lab Program no 8

1 class MultiThreadDemo {
2 public static void main(String args[]) {
3 CollegeThread cseThread = new CollegeThread();
4 cseThread.start();
5 CSEThread cseThread2 = new CSEThread();
6 cseThread2.start();
7 }
8 }
9

Output:

BMS College of Engineering
 CSE
 CSE
 BMS College of Engineering
 CSE
 CSE
 BMS College of Engineering
 CSE
 CSE
 Thread interrupted : sleep interrupted
 Thread interrupted : sleep interrupted

```

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

```

```

class CSEThread extends Thread {
    public void run() {
        try {
            while (true) {

```

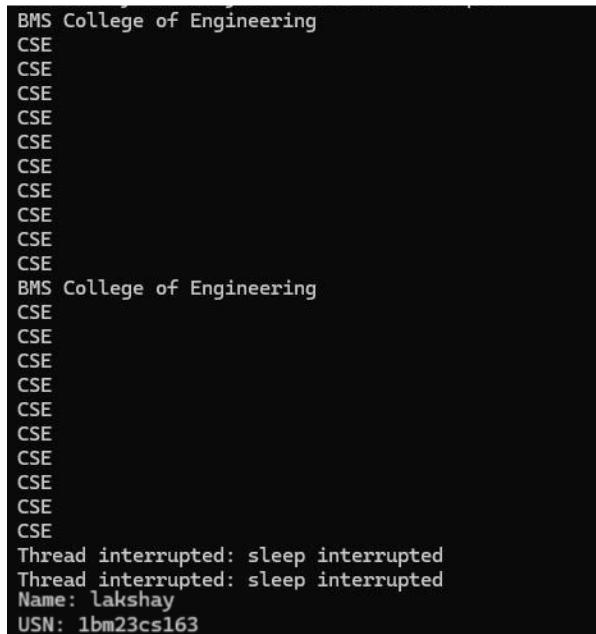
```

        System.out.println("CSE");
        Thread.sleep(2000);
    } catch (InterruptedException e) {
        System.err.println("CSEThread interrupted.");
    }
}

public class Main {
    public static void main(String[] args) {
        BMSCollegeThread bmsThread = new BMSCollegeThread();
        CSEThread cseThread = new CSEThread();

        bmsThread.start();
        cseThread.start();
    }
}

```



A terminal window displaying the output of a Java application. The application consists of two threads: BMSCollegeThread and CSEThread. Both threads print "CSE" repeatedly. The CSEThread also prints its name and USN at the end. The BMSCollegeThread prints an interrupt message.

```

BMS College of Engineering
CSE
BMS College of Engineering
CSE
Thread interrupted: sleep interrupted
Thread interrupted: sleep interrupted
Name: lakshay
USN: 1bm23cs163

```

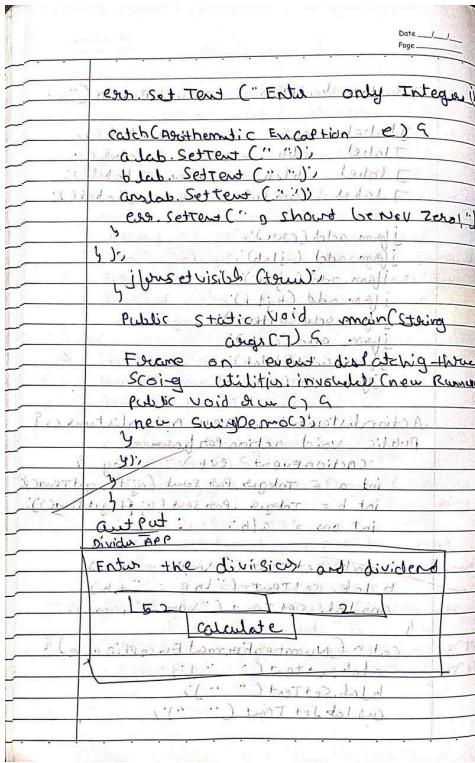
Program 9

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 ArithmeticException. Display the exception in a message dialog box.

Algorithm:-

<p>LAB Program - 9</p> <pre> a. WAP that creates an interface to perform integer division. The user enters 2 numbers in the two fields, Num1 & Num2. The division of Num1 & Num2 is displayed in the result field when the divide button is clicked. If num1 or num2 are not integers, the program throws an arithmetic exception. Display the exception in a message dialog box. </pre> <pre> import javax.swing.*; import java.awt.*; import java.awt.event.*; import java.util.*; class SwingDemo { JFrame jfrm = new JFrame ("Divide APP"); jfrm.setSize(275, 150); jfrm.setLayout(new FlowLayout()); jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); JLabel jlab = new JLabel ("Enter the divisor & dividend:"); JTextField aift = new JTextField("A"); JTextField bift = new JTextField("B"); JButton button = new JButton("Divide"); button.addActionListener(new ActionListener() { public void actionPerformed(ActionEvent evt) { int a = Integer.parseInt(aift.getText()); int b = Integer.parseInt(bift.getText()); int ans = a/b; aLab.setText("A = " + a); bLab.setText("B = " + b); ansLab.setText("Ans = " + ans); } }); catch (NumberFormatException e) { aLab.setText("A = "); bLab.setText("B = "); ansLab.setText("Ans = "); } } </pre>	<p>Date / / Page / /</p> <pre> JButton button = new JButton(); JLabel aLab = new JLabel(); JLabel bLab = new JLabel(); JLabel ansLab = new JLabel(); button.addActionListener(new ActionListener() { public void actionPerformed(ActionEvent evt) { int a = Integer.parseInt(aift.getText()); int b = Integer.parseInt(bift.getText()); int ans = a/b; aLab.setText("A = " + a); bLab.setText("B = " + b); ansLab.setText("Ans = " + ans); } }); catch (NumberFormatException e) { aLab.setText("A = "); bLab.setText("B = "); ansLab.setText("Ans = "); } </pre>
---	---



Code:-

```

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

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

        JFrame frame = new JFrame("Integer Division");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 200);
        frame.setLayout(new GridLayout(4, 2, 10, 10));

        JLabel labelNum1 = new JLabel("Num1:");
        JTextField textNum1 = new JTextField();
        JLabel labelNum2 = new JLabel("Num2:");
        
```

```
JTextField textNum2 = new JTextField();
JLabel labelResult = new JLabel("Result:");
JTextField textResult = new JTextField();
textResult.setEditable(false);
JButton buttonDivide = new JButton("Divide");
```

```
frame.add(labelNum1);
frame.add(textNum1);
frame.add(labelNum2);
frame.add(textNum2);
frame.add(labelResult);
frame.add(textResult);
frame.add(buttonDivide);
```

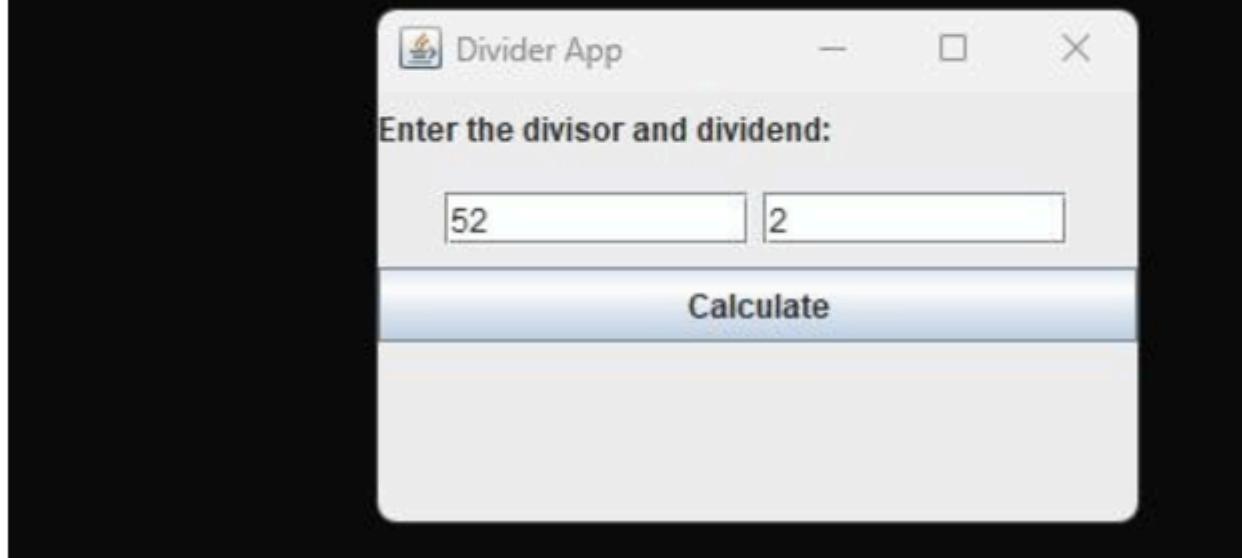
```
buttonDivide.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            int num1 = Integer.parseInt(textNum1.getText());
            int num2 = Integer.parseInt(textNum2.getText());

            if (num2 == 0) {
                throw new ArithmeticException("Division by zero is not
allowed");
            }

            int result = num1 / num2;
            textResult.setText(String.valueOf(result));
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(frame, "Please enter valid
integers.", "Input Error", JOptionPane.ERROR_MESSAGE);
        } catch (ArithmaticException ex) {
            JOptionPane.showMessageDialog(frame, ex.getMessage(),
"Arithmatic Error", JOptionPane.ERROR_MESSAGE);
        }
    }
})
```

```
});  
frame.setVisible(true);  
}  
}  
}
```

```
javac SwingDemo.java  
java SwingDemo
```



Program:- 10

Demonstrate Inter process Communication and deadlock

Algorithm:-

LAB Program -10

Date / / Page / /

- Demonstrate Inter Process Communication and deadlock

A) Deadlock

```

    package Lab1;
    import java.util.*;
    import java.lang.*;

    class A {
        synchronized void bar(A a) {
            String name = Thread.currentThread();
            System.out.println(name + " entered A");
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                System.out.println("A interrupted");
            }
            System.out.println(name + " trying to call B last()");
            synchronized void last() {
                System.out.println("Inside A-last");
            }
        }
    }

    class B {
        synchronized void bar(B b) {
            String name = Thread.currentThread();
            System.out.println(name + " entered B");
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                System.out.println("B interrupted");
            }
            System.out.println(name + " trying to call A last()");
            synchronized void last() {
                System.out.println("Inside B-last");
            }
        }
    }
}
  
```

LAB Program -10

Date / / Page / /

```

    package Lab1;
    import java.util.*;
    import java.lang.*;

    class Deadlock implements Runnable {
        Thread t;
        public Deadlock() {
            t = new Thread(this, "Racing Thread");
            t.start();
        }
        @Override
        public void run() {
            synchronized void bar(A a) {
                String name = Thread.currentThread();
                System.out.println(name + " entered A");
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    System.out.println("A interrupted");
                }
                System.out.println(name + " trying to call B last()");
                synchronized void last() {
                    System.out.println("Inside A-last");
                }
            }
            synchronized void bar(B b) {
                String name = Thread.currentThread();
                System.out.println(name + " entered B");
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    System.out.println("B interrupted");
                }
                System.out.println(name + " trying to call A last()");
                synchronized void last() {
                    System.out.println("Inside B-last");
                }
            }
        }
    }
}
  
```

```

public static void main() {
    String[] args = new String[1];
    new Deadlock();
}

class A {
    synchronized void foo() {
        String name = Thread.currentThread();
        System.out.println(name + " entered A");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println("A interrupted");
        }
        System.out.println(name + " trying to call B last()");
        synchronized void last() {
            System.out.println("Inside A-last");
        }
    }
}

class B {
    synchronized void foo() {
        String name = Thread.currentThread();
        System.out.println(name + " entered B");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println("B interrupted");
        }
        System.out.println(name + " trying to call A last()");
        synchronized void last() {
            System.out.println("Inside B-last");
        }
    }
}
  
```

Thread. sleep (1000);
 y. catch (InterruptedException ex);
 System.out.println ("A Interruped");
 y.
 System.out.println ("name + " trying
 Call B. last (););
 b. last ();
 y.
 void last () {
 System.out.println ("Inside A last");
 }
 y. A. last ();
 y. A. last ();
 Class B extends A {
 synchronized void bar (Lab2.A)
 String name = Thread.currentThread().getName();
 System.out.println ("name + " trying to call B. last");
 b. last ();
 y.
 void last () {
 System.out.println ("In Side A");
 }
 y.
 Class B {
 }

synchronized void bar (Lab2.A) {
 String name = Thread.currentThread().getName();
 System.out.println ("name + " entered
 b. last ");
 }
 try {
 Thread.sleep (1000);
 }
 catch (InterruptedException ex) {
 System.out.println ("B Interruped");
 }
 System.out.println ("name + " trying to
 call a. last (););
 a. last ();
 y.
 void last () {
 System.out.println ("Inside B last");
 }
 y.
 Class B implements Runnable {
 public void run () {
 Lab2.A a = new Lab2.A ();
 Lab2.B b = new Lab2.B ();
 IPS ips = new IPS ();
 Thread currentThread = Thread.currentThread();
 currentThread.setName ("Main Thread");
 Thread t = new Thread (this,
 "Racing Thread");
 t.start ();
 a. sleep ();
 }

System.out.println ("Back in
 main thread");
 y.
 Class B {
 }

@Override
 public void run () {
 b. bar ();
 y.
 public static void main (String args) {
 Thread t = new Thread (new IPS ());
 t.start ();
 3. if ("Racing Thread")
 authPut:
 main Thread (currentThread, "Racing Thread inCurrent B bar");
 main Thread trying to call B. last()
 Racing Thread trying to call A. last
 Thread B. last
 Thread A. last ();
 Back in Main Thread
 main Thread is a SubThread
 call back of a class
 y.
 main Thread (currentThread, "Racing Thread inCurrent B bar");
 if ("Racing Thread")
 authPut:
 main Thread (currentThread, "Racing Thread inCurrent B bar");
 if ("Racing Thread")
 authPut:
 main Thread (currentThread, "Racing Thread inCurrent B bar");

Code:-

PART A

```
class Resource1 {  
    synchronized void lock(Resource2 res2) {  
        System.out.println(Thread.currentThread().getName() + " locked  
Resource1");  
  
        try {  
            Thread.sleep(100);  
        } catch (InterruptedException e) {  
            System.out.println(e);  
        }  
  
        System.out.println(Thread.currentThread().getName() + " trying to lock  
Resource2...");  
        res2.method2();  
    }  
  
    synchronized void method1() {  
        System.out.println(Thread.currentThread().getName() + " is working with  
Resource1");  
    }  
}  
  
class Resource2 {  
    synchronized void lock(Resource1 res1) {  
        System.out.println(Thread.currentThread().getName() + " locked  
Resource2");  
  
        try {  
            Thread.sleep(100);  
        } catch (InterruptedException e) {  
            System.out.println(e);  
        }  
  
        System.out.println(Thread.currentThread().getName() + " trying to lock  
Resource1...");  
    }  
}
```

```

        res1.method1();
    }

    synchronized void method2() {
        System.out.println(Thread.currentThread().getName() + " is working with
Resource2");
    }
}

public class DeadlockDemo {
    public static void main(String[] args) {
        Resource1 res1 = new Resource1();
        Resource2 res2 = new Resource2();
        Thread t1 = new Thread(() -> res1.lock(res2), "Thread-1");
        Thread t2 = new Thread(() -> res2.lock(res1), "Thread-2");

        t1.start();
        t2.start();
    }
}

```

PART B

```

import java.util.concurrent.locks.Lock;
import java.util.concurrent.locks.ReentrantLock;

class SafeResource1 {
    private final Lock lock = new ReentrantLock();
    public boolean tryLockBoth(SafeResource1 otherResource) {
        while (true) {
            boolean gotFirstLock = lock.tryLock();
            resource's lock
            boolean gotSecondLock = otherResource.tryLock(); the other resource's
lock

            if (gotFirstLock && gotSecondLock) {
                System.out.println(Thread.currentThread().getName() + " acquired

```

```

both locks");
        return true;
    }

    if (gotFirstLock) lock.unlock();
    if (gotSecondLock) otherResource.unlock();
}
}

public boolean tryLock() {
    return lock.tryLock();
}

public void unlock() {
    lock.unlock();
}
}

public class DeadlockResolved {
    public static void main(String[] args) {
        SafeResource1 res1 = new SafeResource1();
        SafeResource1 res2 = new SafeResource1();

        Thread t1 = new Thread(() -> {
            if (res1.tryLockBoth(res2)) {
                System.out.println("Thread-1 completed safely");
                res1.unlock();
                res2.unlock();
            }
        }, "Thread-1");

        Thread t2 = new Thread(() -> {
            if (res2.tryLockBoth(res1)) {
                System.out.println("Thread-2 completed safely");
                res2.unlock();
                res1.unlock();
            }
        }, "Thread-2");
    }
}

```

```
    t1.start();
    t2.start();
}
}
```

OutPut : A

```
Put: 0
Intimate Consumer

Producer waiting

Got: 0
Intimate Producer

Put: 1
Intimate Consumer

Producer waiting
Consumed: 0
Got: 1
Intimate Producer

Consumed: 1
Put: 2

Intimate Consumer

Producer waiting
Got: 2
Intimate Producer

Consumed: 2
Put: 3

Intimate Consumer

Producer waiting
Got: 3
```

OutPut B:-

```
MainThread entered A.foo
RacingThread entered B.bar
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
Name: lakshay
USN: 1bm23cs163
```