

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT on **Object Oriented Java Programming (23CS3PCOOJ)**

Submitted by

Anji Lakshmi (24BECS421)

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 **Anji Lakshmi (24BECS421)**, who is bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Dr. Prasad G R Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
---	---

Index

Sl. No.	Date	Experiment Title	Page No.
1	09-10-2024	Quadratic Equation	4-6
2	16-10-2024	Array of objects	7-11
3	16-10-2024	Constructors in java	12-16
4	23-10-2024	Abstract class in java	17-21
5	13-11-2024	Inheritance in java	22-27
6	13-11-2024	Packages in java	28-33
7	20-11-2024	Exception Handling	34-37
8	27-11-2024	Threads in java	38-40
9	27-11-2024	User interface	41-44
10	27-11-2024	Inter process communication and deadlock	46-53

Github Link:

<https://github.com/AnjiLakshmi234/java-lab.git>

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.

Lab 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 this discriminant $b^2 - 4ac$ is negative; display a message stating that there are no real solutions.

```
import java.util.Scanner;
class QuadraticEquationSolver {
    public static void main (String[] args) {
        Scanner scanner = new Scanner (System.in);
        System.out.print ("Enter coefficient a: ");
        double a = scanner.nextDouble();
        System.out.print ("Enter coefficient b: ");
        double b = scanner.nextDouble();
        System.out.print ("Enter coefficient c: ");
        double c = scanner.nextDouble();
        double discriminant = b*b - 4*a*c;
        if (discriminant > 0) {
            double root1 = (-b + Math.sqrt (discriminant)) / (2*a);
            double root2 = (-b - Math.sqrt (discriminant)) / (2*a);
            System.out.println ("There are two real solutions: x1 = " + root1 +
                ", x2 = " + root2);
        } else if (discriminant == 0) {
            double root = -b / (2*a);
            System.out.println ("There is one real solution: x = " + root);
        } else {
            System.out.println ("There are no real solutions.");
        }
        scanner.close();
    }
}
```

output : $(a+b)(a-b)$

Starting set of variables for the class QuadraticEquationSolver
Value of coefficient a: 1
Value of coefficient b: 2
Value of coefficient c: 1
→ There is one real solution: x = 1.00

① Enter coefficient a: 1
Enter coefficient b: -3
Enter coefficient c: 2
There are two real solutions: x1 = 2.00, x2 = 1.00

② Enter coefficient a: 2
Enter coefficient b: 5
Enter coefficient c: -2
There are two real solutions: x1 = -0.50, x2 = -2.00

Code:

```
import java.util.Scanner;

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

        System.out.println("Name: Anji Lakshmi");
        System.out.println("USN: 24BECS421");

        Scanner scanner = new Scanner(System.in);

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

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

        if (discriminant > 0) {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.printf("There are two real solutions: x1 = %.2f, x2 = %.2f\n", root1, root2);
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.printf("There is one real solution: x = %.2f\n", root);
        } else {
            System.out.println("There are no real solutions.");
        }

        scanner.close();
    }
}
```

output:

```
C:\Users\anjil\Desktop\java>java QuadraticEquationSolver
Name: Anji Lakshmi
USN: 24BECS421
Enter coefficient a: 1
Enter coefficient b: -2
Enter coefficient c: 1
There is one real solution: x = 1.00

C:\Users\anjil\Desktop\java>java QuadraticEquationSolver
Name: Anji Lakshmi
USN: 24BECS421
Enter coefficient a: 1
Enter coefficient b: -3
Enter coefficient c: 2
There are two real solutions: x1 = 2.00, x2 = 1.00

C:\Users\anjil\Desktop\java>java QuadraticEquationSolver
Name: Anji Lakshmi
USN: 24BECS421
Enter coefficient a: 1
Enter coefficient b: 0
Enter coefficient c: 1
There are no real solutions.
```

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.

Week 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.

```
→ import java.util.Scanner;
class student {
    String usn;
    String name;
    int[] credits;
    int[] marks;
    int n;
    void acceptDetails() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter usn: ");
        usn = sc.nextLine();
        System.out.print("Enter name: ");
        name = sc.nextLine();
        System.out.print("Enter number of subjects: ");
        n = sc.nextInt();
        credits = new int[n];
        marks = new int[n];
        for (int i=0; i<n; i++) {
            System.out.print("Enter credits for subject " + (i+1) + ": ");
            credits[i] = sc.nextInt();
            System.out.print("Enter marks for subject " + (i+1) + ": ");
            marks[i] = sc.nextInt();
        }
    }
    void displayDetails() {
        System.out.println("\nStudent Details:");
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Subject wise details:");
        for (int i=0; i<n; i++) {
            System.out.println("Subject " + (i+1) + "; Credits: " + credits[i] +
                               ", Marks: " + marks[i]);
        }
    }
    double calculateSGPA() {
        int totalCredits = 0;
        double totalPoints = 0.0;
        for (int i=0; i<n; i++) {
            int gradePoint = calculateGradePoint(marks[i]);
            totalCredits += credits[i];
            totalPoints += gradePoint * credits[i];
        }
        return totalPoints / totalCredits;
    }
    int calculateGradePoint(int marks) {
        if (marks >= 90) return 10;
        else if (marks >= 80) return 9;
        else if (marks >= 70) return 8;
        else if (marks >= 60) return 7;
        else if (marks >= 50) return 6;
    }
}
```

```

        else if (marks >= 40) return 5;
    else return 0;
}

public static void main (String [] args) {
    Student s = new Student ("");
    s.acceptDetails ();
    s.displayDetails ();
    double sgpa = s.calculateGPA ();
    System.out.println ("SGPA : " + sgpa);
}

```

Output

```

Enter USN : 24BEC5421
Enter name : Anji Lakshmi
Enter number of subjects : 6
Enter credits for subject 1 : 3
Enter marks for subject 1 : 90
Enter marks for subject 2 : 30
Enter marks for subject 2 : 80
Enter credits for subject 3 : 8
Enter marks for subject 3 : 90
Enter credits for subject 4 : 3
Enter marks for subject 4 : 75
Enter credits for subject 5 : 3
Enter marks for subject 5 : 70
Enter credits for subject 6 : 3
Enter marks for subject 6 : 60
SGPA : 8.629

```

Code :

```
import java.util.Scanner;

class Student {
    String usn;
    String name;
    int[] credits;
    int[] marks;
    int n; // Number of subjects

    void acceptDetails() {
        Scanner sc = new Scanner(System.in);

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

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

        System.out.print("Enter number of subjects: ");
        n = sc.nextInt();

        credits = new int[n];
        marks = new int[n];

        for (int i = 0; i < n; i++) {
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            credits[i] = sc.nextInt();

            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            marks[i] = sc.nextInt();
        }
    }

    void displayDetails() {
        System.out.println("\nStudent Details:");
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Subject-wise details:");

        for (int i = 0; i < n; i++) {
            System.out.println("Subject " + (i + 1) + ": Credits = " + credits[i] + ", Marks = " + marks[i]);
        }
    }
}
```

```

}

double calculateSGPA() {
    int totalCredits = 0;
    double totalPoints = 0.0;

    for (int i = 0; i < n; i++) {
        int gradePoint = calculateGradePoint(marks[i]);
        totalCredits += credits[i];
        totalPoints += gradePoint * credits[i];
    }

    return totalPoints / totalCredits;
}

int calculateGradePoint(int marks) {
    if (marks >= 90) return 10;
    else if (marks >= 80) return 9;
    else if (marks >= 70) return 8;
    else if (marks >= 60) return 7;
    else if (marks >= 50) return 6;
    else if (marks >= 40) return 5;
    else return 0; // Fail
}

public static void main(String[] args) {
    Student s = new Student();
    s.acceptDetails();
    s.displayDetails();

    double sgpa = s.calculateSGPA();
    System.out.println("SGPA: " + sgpa);
}
}

```

Output:

```
C:\Users\bmsce\Desktop\oop>java Student
Enter USN: Ashwini
Enter Name: 24BECS430
Enter number of subjects: 9
Enter credits for subject 1: 3
Enter marks for subject 1: 80
Enter credits for subject 2: 3
Enter marks for subject 2: 78
Enter credits for subject 3: 2
Enter marks for subject 3: 78
Enter credits for subject 4: 3
Enter marks for subject 4: 99
Enter credits for subject 5: 4
Enter marks for subject 5: 80
Enter credits for subject 6: 3
Enter marks for subject 6: 98
Enter credits for subject 7: 3
Enter marks for subject 7: 89
Enter credits for subject 8: 3
Enter marks for subject 8: 79
Enter credits for subject 9: 3
Enter marks for subject 9: 80
```

Student Details:

USN: Ashwini
Name: 24BECS430

Subject-wise details:

Subject 1: Credits = 3, Marks = 80
Subject 2: Credits = 3, Marks = 78
Subject 3: Credits = 2, Marks = 78
Subject 4: Credits = 3, Marks = 99
Subject 5: Credits = 4, Marks = 80
Subject 6: Credits = 3, Marks = 98
Subject 7: Credits = 3, Marks = 89
Subject 8: Credits = 3, Marks = 79
Subject 9: Credits = 3, Marks = 80

SGPA: 8.925925925925926

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

```
C:\Users\bmsce\Desktop\oop>javac Student.java
```

```
C:\Users\bmsce\Desktop\oop>java Student
Enter USN: 24BECS421
Enter Name: Anji Lakshmi
Enter number of subjects: 9
Enter credits for subject 1: 3
Enter marks for subject 1: 90
Enter credits for subject 2: 3
Enter marks for subject 2: 80
Enter credits for subject 3: 2
Enter marks for subject 3: 80
Enter credits for subject 4: 3
Enter marks for subject 4: 75
Enter credits for subject 5: 4
Enter marks for subject 5: 70
Enter credits for subject 6: 3
Enter marks for subject 6: 66
Enter credits for subject 7: 3
Enter marks for subject 7: 98
Enter credits for subject 8: 3
Enter marks for subject 8: 67
Enter credits for subject 9: 3
Enter marks for subject 9: 90
```

Student Details:

USN: 24BECS421
Name: Anji Lakshmi

Subject-wise details:

Subject 1: Credits = 3, Marks = 90
Subject 2: Credits = 3, Marks = 80
Subject 3: Credits = 2, Marks = 80
Subject 4: Credits = 3, Marks = 75
Subject 5: Credits = 4, Marks = 70
Subject 6: Credits = 3, Marks = 66
Subject 7: Credits = 3, Marks = 98
Subject 8: Credits = 3, Marks = 67
Subject 9: Credits = 3, Marks = 90

SGPA: 8.62962962962963

```
C:\Users\bmsce\Desktop\oop>
```

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.

Week : 3
Construction Class

```
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 getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public int getNumPages() {
        return numPages;
    }

    public void setName (String name) {
        this.name = name;
    }

    public void setAuthor (String author) {
        this.author = author;
    }

    public void setPrice (double price) {
        this.price = price;
    }

    public void setNumPages (int numPages) {
        this.numPages = numPages;
    }

    public String toString () {
        return "Book Name : " + name + "\n Author : " + author
               + "\n price : " + price + "\n number of pages : " + numPages;
    }
}

public static void main (String [] args) {
    Scanner scanner = new Scanner (System.in);
    System.out.println ("Enter the number of books you want to create");
    int n = scanner.nextInt ();
    scanner.nextLine ();
    Book [] books = new Book [n];
    for (int i=0; i<n; i++) {
        System.out.println ("Enter details for book " + (i+1));
        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: ");
        double price = scanner.nextDouble ();
        System.out.print ("Enter number of pages: ");
        int numPages = scanner.nextInt ();
        scanner.nextLine ();
        books[i] = new Book (name, author, price, numPages);
    }
}
```

```

1. books[i] = new Book(name, author, price, numPages);
2. System.out.println("Book Details:");
3. for (int i=0; i<n; i++) {
    System.out.println("Details of Book " + (i+1));
    System.out.println(books[i].toString());
}
4. scanner.close();

```

Output :

Name : Anji Lakshmi USN : 24BEC342
Enter the number of books : 2
Enter details for book 1:
Name : SDM
Author : D. S. Chandrasekhariah
price : 200
Number of pages : 300
Enter the number of books : 2:
Name : C.A.
Author : David A
Price : 150
Number of pages : 400

Code :

```

import java.util.ArrayList;
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 getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public int getNumPages() {
        return numPages;
    }

    public void setNumPages(int numPages) {
        this.numPages = numPages;
    }

    @Override
    public String toString() {
        return "Book{" +
            "name=\"" + name + '\"' +
            ", author=\"" + author + '\"' +
            ", price=" + price +
            ", numPages=" + numPages +
            '}';
    }
}

```

```

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

        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();
        scanner.nextLine();

        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: ");
            double price = scanner.nextDouble();

            System.out.print("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 all books:");
        for (Book book : books) {
            System.out.println(book);
        }

        scanner.close();
    }
}

```

Output:

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

C:\Users\BMSCE\Desktop\java>javac BookStore.java

C:\Users\BMSCE\Desktop\java>java BookStore
Name:Anji LakshmiUSN:24BECS421
Enter the number of books: 6
Enter details for book 1:
Name: SDM
Author: D. S. Chandrasekharaiah
Price: 200
Number of pages: 400
Enter details for book 2:
Name: COA
Author: David A
Price: 150
Number of pages: 300
Enter details for book 3:
Name: OOD
Author: Y. Daniel Liang
Price: 220
Number of pages: 400
Enter details for book 4:
Name: LD
Author: R D Sudhaker Samuel
Price: 150
Number of pages: 300
Enter details for book 5:
Name: DS
Author: lakshmi
Price: 200
Number of pages: 450
Enter details for book 6:
Name: DBMS
Author: ashwini
Price: 220
Number of pages: 430

Details of all books:
Book{name='SDM', author='D. S. Chandrasekharaiah', price=200.0, numPages=400}
Book{name='COA', author='David A', price=150.0, numPages=300}
Book{name='OOD', author='Y. Daniel Liang', price=220.0, numPages=400}
Book{name='LD', author='R D Sudhaker Samuel', price=150.0, numPages=300}
Book{name='DS', author='lakshmi', price=200.0, numPages=450}
Book{name='DBMS', author='ashwini', price=220.0, numPages=430}

C:\Users\BMSCE\Desktop\java>
```

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.

WEEK : 4
Abstract

```
import java.util.Scanner;
abstract class Shape {
    int dimension1;
    int dimension2;
    abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle (int length, int width) {
        this.dimension1 = length;
        this.dimension2 = width;
    }

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

class Triangle extends Shape {
    public Triangle (int base, int height) {
        this.dimension1 = base;
        this.dimension2 = height;
    }

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

class Circle extends Shape {
    private final double pi = 3.14159;
    int radius;
    int dimension1 = radius;
    int dimension2 = 0;

    void printArea() {
        double area = pi * dimension1 * dimension2;
        System.out.println ("Circle Area : " + area);
    }
}

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

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

        System.out.print ("Enter radius of circle : ");
        int modulus = scanner.nextInt ();
    }
}
```

```

        Circle circle = new Circle(5);
        circle.printArea();
    }

    scanner.close();
}

}

```

Output :

```

Enter length of rectangle : 20
Enter width of rectangle : 6
Rectangle area : 120
Enter base of triangle : 2
Enter height of triangle : 3
Triangle area : 3.0
Enter radius of circle : 3
Circle area : 28.2743338823

```

~~Opposite side of triangle is 4.0~~

~~Perimeter of triangle is 10.0~~

~~Perimeter of rectangle is 52.0~~

~~Perimeter of circle is 18.85~~

~~Opposite side of triangle is 4.0~~

~~Perimeter of rectangle is 52.0~~

~~Perimeter of circle is 18.85~~

Code :

```
import java.util.Scanner;

abstract class Shape {
    int dimension1;
    int dimension2;

    abstract void printArea();
}

class Rectangle extends Shape {

    public Rectangle(int length, int width) {
        this.dimension1 = length;
        this.dimension2 = width;
    }

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

class Triangle extends Shape {

    public Triangle(int base, int height) {
        this.dimension1 = base;
        this.dimension2 = height;
    }

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

class Circle extends Shape {
    private final double pi = 3.14159;
```

```

public Circle(int radius) {
    this.dimension1 = radius;
    this.dimension2 = 0;
}

void printArea() {
    double area = pi * dimension1 * dimension1;
    System.out.println("Circle Area: " + area);
}
}

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

        System.out.println("Name:Anji Lakshmi , USN:24BECS421");
        System.out.print("Enter length of rectangle: ");
        int length = scanner.nextInt();
        System.out.print("Enter width of rectangle: ");
        int width = scanner.nextInt();
        Rectangle rectangle = new Rectangle(length, width);
        rectangle.printArea();

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

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

        scanner.close();
    }
}

```

Output :

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

C:\Users\BMSCE\Desktop>java Main
Name:Anji Lakshmi , USN:24BECS421
Enter length of rectangle: 2 6
Enter width of rectangle: Rectangle Area: 12
Enter base of triangle: 2 3
Enter height of triangle: Triangle Area: 3.0
Enter radius of circle: 3
Circle Area: 28.274309999999996

C:\Users\BMSCE\Desktop>java Main
Name:Anji Lakshmi , USN:24BECS421
Enter length of rectangle: 4 8
Enter width of rectangle: Rectangle Area: 32
Enter base of triangle: 2 9
Enter height of triangle: Triangle Area: 9.0
Enter radius of circle: 5
Circle Area: 78.53975

C:\Users\BMSCE\Desktop>
```

Program 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

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

```

Week : 5: Creating object oriented
        / OOPs
        / Class & Object

import java.util.Scanner;
class Account {
    private String customername;
    private 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: " + amount);
    }
    public void displaybalance () {
        System.out.println ("Balance amount: " + balance);
    }
    public void withdraw (double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println ("Withdraw amount: " + amount);
        } else {
            System.out.println ("Insufficient balance for withdrawal!");
        }
    }
}

protected double getbalance () {
    return balance;
}

class Savacct (String customername, String accountnumber, double interestrate) {
    super (customername, accountnumber);
    this.interestrate = interestrate;
    public void computeanddepositinterest () {
        double currentbalance = getbalance ();
        double interest = currentbalance * interestrate / 100;
        deposit (interest);
        System.out.println ("Interest deposited: " + interest);
    }
}

class Curacct (String customername, String accountnumber, double minimumbalance, double servicecharge) {
    super (customername, accountnumber);
    this.minimumbalance = minimumbalance;
    this.servicecharge = servicecharge;
    public void withdraw (double amount) {
        if (getBalance () < amount + minimumbalance) {
            System.out.println ("Service charge imposed: " + servicecharge);
            deposit (-servicecharge);
            System.out.println ("Insufficient balance");
        } else {
            super.withdraw (amount);
        }
    }
}

```

```

public class Bank {
    public static void main (String[] args) {
        Scanner scanner = new Scanner (System.in);
        System.out.println ("Enter customer name for savings account:");
        String savingsCustomerName = scanner.nextLine();
        System.out.println ("Enter account number for savings account:");
        System.out.println ("Enter interest rate for savings account:");
        double interestRate = scanner.nextDouble();
        SavingsAccount savingsAccount = new Savings (savingsCustomerName, savingsAccountNumber, interestRate);
        savingsAccount.deposit (1000);
        savingsAccount.computeAndDepositInterest ();
        savingsAccount.displayBalance ();
        System.out.println ("Enter amount to withdraw from savings Account");
        double withdrawAmount = scanner.nextDouble();
        savingsAccount.withdraw (withdrawAmount);
        savingsAccount.displayBalance ();
        scanner.nextLine();
        System.out.println ("Enter customer name for Current Account:");
        String currentCustomerName = scanner.nextLine();
        System.out.println ("Enter account number for Current Account:");
        String currentAccountNumber = scanner.nextLine();
        System.out.println ("Enter minimum balance for Current Account:");
        System.out.println ("Enter service charge for Current account:");
        double minimumBalance = scanner.nextDouble();
        double serviceCharge = scanner.nextDouble();
        CurrentAccount currentAccount = new Current (currentCustomerName, currentAccountNumber, minimumBalance, serviceCharge);
        currentAccount.withdraw (currentWithdrawAmount);
        currentAccount.displayBalance ();
        System.out.println ("Enter amount to withdraw from current account");
        double currentWithdrawAmount = scanner.nextDouble();
        CurrentAccount.currentWithdraw (currentWithdrawAmount);
    }
}

```

scanner.close();

}

Output

Saving account

```

Balance : 1000.00
Deposited : 100.00
Balance : 1100.00
Interest of 75.0 has been added
Balance : 1175.0
Withdraw : 300.0
Balance : 875.0

```

Current account

```

Balance : 1200.50
Deposited : 12000.50
Balance : 112500.00

```

Code :

```
import java.util.Scanner;

class Account {
    private String customerName;
    private 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: " + amount);
    }

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

    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdraw amount: " + amount);
        } else {
            System.out.println("Insufficient balance for withdrawal!");
        }
    }

    protected double getBalance() {
        return balance;
    }
}

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 currentBalance = getBalance();
        double interest = currentBalance * interestRate / 100;
        deposit(interest);
        System.out.println("Interest deposited: " + interest);
    }
}

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

    public CurAcct(String customerName, String accountNumber, double minimumBalance, double serviceCharge) {
        super(customerName, accountNumber);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    public void withdraw(double amount) {
        if (getBalance() - amount < minimumBalance) {
            System.out.println("Service charge imposed: " + serviceCharge);
            deposit(-serviceCharge);
            System.out.println("Insufficient balance.");
        } else {
            super.withdraw(amount);
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("NAME : ANJI LAKSHMI");

        System.out.println("USN NO : 24BECS421");

        System.out.println("Enter customer name for Savings Account:");
        String savingsCustomerName = scanner.nextLine();
        System.out.println("Enter account number for Savings Account:");
        String savingsAccountNumber = scanner.nextLine();
        System.out.println("Enter interest rate for Savings Account:");
        double interestRate = scanner.nextDouble();

        SavAcct savingsAccount = new SavAcct(savingsCustomerName, savingsAccountNumber,
interestRate);
    }
}

```

```

savingsAccount.deposit(1000);
savingsAccount.computeAndDepositInterest();
savingsAccount.displayBalance();

System.out.println("Enter amount to withdraw from Savings Account:");
double withdrawAmount = scanner.nextDouble();
savingsAccount.withdraw(withdrawAmount);
savingsAccount.displayBalance();

scanner.nextLine();
System.out.println("Enter customer name for Current Account:");
String currentCustomerName = scanner.nextLine();
System.out.println("Enter account number for Current Account:");
String currentAccountNumber = scanner.nextLine();
System.out.println("Enter minimum balance for Current Account:");
double minimumBalance = scanner.nextDouble();
System.out.println("Enter service charge for Current Account:");
double serviceCharge = scanner.nextDouble();

CurAcct currentAccount = new CurAcct(currentCustomerName, currentAccountNumber,
minimumBalance, serviceCharge);
currentAccount.deposit(2000);
currentAccount.displayBalance();

System.out.println("Enter amount to withdraw from Current Account:");
double currentWithdrawAmount = scanner.nextDouble();
currentAccount.withdraw(currentWithdrawAmount);
currentAccount.displayBalance();

System.out.println("Enter amount to withdraw from Current Account (may incur service
charge):");
currentWithdrawAmount = scanner.nextDouble();
currentAccount.withdraw(currentWithdrawAmount);
currentAccount.displayBalance();

scanner.close();
}
}

```

Output :

```
C:\Users\Admin\Sam\24BECs421>java Bank
NAME : ANJI LAKSHMI
USN NO : 24BECS421
Enter customer name for Savings Account:
Anji Lakshmi
Enter account number for Savings Account:
67984472
Enter interest rate for Savings Account:
2
Deposited amount: 1000.0
Deposited amount: 20.0
Interest deposited: 20.0
Balance amount: 1020.0
Enter amount to withdraw from Savings Account:
500
Withdraw amount: 500.0
Balance amount: 520.0
Enter customer name for Current Account:
Anji Lakshmi
Enter account number for Current Account:
67984472
Enter minimum balance for Current Account:
520
Enter service charge for Current Account:
20
Deposited amount: 2000.0
Balance amount: 2000.0
Enter amount to withdraw from Current Account:
200
Withdraw amount: 200.0
Balance amount: 1800.0
Enter amount to withdraw from Current Account (may incur service charge):
200
Withdraw amount: 200.0
Balance amount: 1600.0
```

Program 6

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

Week :- 6

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

Folder CIE

```

package CIE;
public class Internals {
    protected int[] internalMarks = new int[5];
    public Internals (int[] marks) {
        if (marks.length == 5) {
            this.internalMarks = marks;
        } else {
            System.out.println ("please provide exactly 5 marks");
        }
    }
    public void displayInternalMarks () {
        System.out.println ("Internal marks : ");
        for (int i=0; i<internalMarks.length; i++) {
            System.out.println ("(" + i + ") : " + internalMarks[i]);
        }
    }
}

```

package CIE;

```

public class Personal {
    protected String usn;
    protected String name;
    protected int sem;
    public Personal (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 - Folder

```

package SEE;
import CIE.Personal;
public class External extends Personal {
    private int[] externalMarks = new int[5];
    public External (String usn, String name, int sem, int[] extend) {
        super (usn, name, sem);
        if (externalMarks.length == 5) {
            this.externalMarks = externalMarks;
        }
    }
}

```

```

        } else {
            System.out.println("please provide exactly 5 external marks");
        }
    }

    public void displayExternalMarks() {
        System.out.println("External marks:");
        for (int i=0; i<externalMarks.length; i++) {
            System.out.println("Outer[" + (i+1) + "]: " + externalMarks[i]);
        }
    }
}

```

→ Main program

```

import CIE.Internal;
import SEE.External;
import java.util.Scanner;
public class StudentMarks {
    public static void main (String[] args) {
        Scanner scanner = new Scanner (System.in);
        System.out.print ("Enter the number of student:");
        int n = scanner.nextInt();
        personal [ ] students = new personal [n];
        Internal [ ] internals = new Internal [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 ("Enter USN : ");
            String usn = scanner.next();

```

```

        System.out.print ("Enter Name : ");
        String name = scanner.nextLine();
        System.out.print ("Enter semester : ");
        int sem = scanner.nextInt();
        System.out.println ("Enter integral marks for 5 subjects");
        int [ ] internalMarks = new int [5];
        for (int j=0; j<5; j++) {
            internalMarks [j] = scanner.nextInt();
        }
        System.out.println ("Enter external marks for 5 subjects");
        int [ ] externalMarks = new int [5];
        for (int j=0; j<5; j++) {
            externalMarks [j] = scanner.nextInt();
        }
        Students [i] = new personal (usn, name, sem);
        internals [i] = new Internal (internalMarks);
        externals [i] = new External (usn, name, sem, externalMarks);
    }

    System.out.println ("Final marks of students:");
    for (int i=0; i<n; i++) {
        Students [i].displayPersonalInfo ();
        internals [i].displayInternalMarks ();
        externals [i].displayExternalMarks ();
        System.out.print ("Final Marks : ");
        for (int j=0; j<5; j++) {
            int finalmark = internals [i].internalMarks [j];
            externals [i].externalMarks [j];
        }
        System.out.println ("\n");
    }
    scanner.close();
}

```

Output:-

Enter number of Students : 1

Enter details for student 1 :

Enter USN : 84BEC5421

Enter name : Anji Lakshmi

Enter semester : 3

Enter internal marks for subjects :

32

33

40

34

76

Enter external marks for 5 subjects :

98

88

66

90

100

Final Marks of students :

USN : 84BEC5421

Semester : 3

Internal marks : 32 33 40 34 76

External marks : 98 88 66 90 100

Final marks : 120 121 106 124 176

Code:

IN THE CIE FOLDER

1

Package CIE;

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

2

package CIE;

```
public class Internals {  
    public int[] internalMarks = new int[5];  
  
    public Internals(int[] marks) {  
        if (marks.length == 5) {  
            System.arraycopy(marks, 0, internalMarks, 0, 5);  
        } else {  
            System.out.println("Error: Exactly 5 marks are required.");  
        }  
    }  
  
    public void displayInternalMarks() {  
        System.out.print("Internal Marks: ");  
        for (int mark : internalMarks) {  
            System.out.print(mark + " ");  
        }  
        System.out.println();  
    }  
}
```

IN THE SEE FOLDER

```
package SEE;

import CIE.Personal;

public class External extends Personal {
    public int[] externalMarks = new int[5];

    public External(String usn, String name, int sem, int[] marks) {
        super(usn, name, sem);
        if (marks.length == 5) {
            System.arraycopy(marks, 0, externalMarks, 0, 5);
        } else {
            System.out.println("Error: Exactly 5 marks are required.");
        }
    }

    public void displayExternalMarks() {
        System.out.print("External Marks: ");
        for (int mark : externalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }
}
```

IN THE MAIN CLASS

```
package SEE;

import CIE.Personal;

public class External extends Personal {
    public int[] externalMarks = new int[5];

    public External(String usn, String name, int sem, int[] marks) {
        super(usn, name, sem);
        if (marks.length == 5) {
            System.arraycopy(marks, 0, externalMarks, 0, 5);
        } else {
            System.out.println("Error: Exactly 5 marks are required.");
        }
    }

    public void displayExternalMarks() {
```

```

        System.out.print("External Marks: ");
        for (int mark : externalMarks) {
            System.out.print(mark + " ");
        }
        System.out.println();
    }
}

```

Output :

```

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

C:\Users\anjil\Desktop\java\week6>javac CIE/Personal.java CIE/Internals.java SEE/External.java Main.java

C:\Users\anjil\Desktop\java\week6>java Main
Enter number of students: 2

Enter details for Student 1:
Enter USN: 24BECS421
Enter Name: Lakshmi
Enter Semester: 3
Enter Internal Marks for 5 subjects:
22
33
40
34
76
Enter External Marks for 5 subjects:
98
88
66
90
100

Enter details for Student 2:
Enter USN: 24BECS401
Enter Name: kusuma
Enter Semester: 3
Enter Internal Marks for 5 subjects:
33
67
90
22
60
Enter External Marks for 5 subjects:
90
88
79
80
98

Final Marks of Students:
USN: 24BECS421
Name: Lakshmi
Semester: 3
Internal Marks: 22 33 40 34 76
External Marks: 98 88 66 90 100
Final Marks: 120 121 106 124 176

USN: 24BECS401
Name: kusuma
Semester: 3
Internal Marks: 33 67 90 22 60
External Marks: 90 88 79 80 98
Final Marks: 123 155 169 102 158

```

Program 7

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

Week 4

Date: 26/11/2024

Demonstrates handling of exceptions in inheritance tree

Create a base class called Father and derived class called as Son extends the base class. In Father class implement the constructor which takes the age and throws the exception wrongAge() when the input is less than 0 (age < 0) in Son class implement a constructor that uses both father and son's age and throws the exception Son's age >= Father's age.

```

import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException (String message) {
        super(message);
    }
}
class SonAgeException extends Exception {
    public SonAgeException (String message) {
        super(message);
    }
}
class Father {
    private int age;
    public Father (int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException ("wrong age");
        }
        this.age = age;
    }
    public int getAge () {
        return age;
    }
}
class Son extends Father {
    private int sonAge;
    public Son (int fatherAge, int sonAge) throws SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException ("son's age cannot be greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge () {
        return sonAge;
    }
}
public class FatherSon {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        while (true) {
            System.out.print ("enter Father's age : ");
            int fatherAge = sc.nextInt ();
            System.out.print ("enter son's age : ");
            System.out.println ("Accepted successfully");
            int sonAge = sc.nextInt ();
            try {
                Son son = new Son (fatherAge, sonAge);
            } catch (WrongAgeException e) {
                System.out.println (e.getMessage ());
            } catch (SonAgeException e) {
                System.out.println (e.getMessage ());
            }
        }
    }
}

```

```
System.out.println("would you like to enter details (y/n)");  
String input = sc.next();  
if (input.equals("n")) {  
    break;  
}
```

Output :-

```
NAME : ANJI LAKSHMI  
USN NO : 24BEC5421  
Enter father's Age : 45  
Enter son's Age : 20  
Accepted successfully  
would you like to enter details (y/n)  
y  
Enter father's Age : 20  
Enter son's Age : 33  
Son's age cannot be greater than or equal to father's age
```

Code:

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

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

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

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

```

System.out.print("Enter Son's Age: ");
int sonAge = sc.nextInt();
try {
    Son son = new Son(fatherAge, sonAge);
    System.out.println("Accepted Successfully");
}
catch (WrongAgeException e) {
    System.out.println(e.getMessage());
}
catch (SonAgeException e) {
    System.out.println(e.getMessage());
}
System.out.println("Would you like to re-enter details (Y/n)");
String input = sc.next();
if (input.equalsIgnoreCase("n")) {
    break;
}
}
}
}
}

```

Output:

```

C:\Users\Admin\Sam\24BECs421>javac FatherSon.java

C:\Users\Admin\Sam\24BECs421>java FatherSon
•
NAME : ANJI LAKSHMI
USN NO : 24BECS421
Enter Father's Age: 45
Enter Son's Age: 20
• Accepted Successfully
Would you like to re-enter details (Y/n)
y
NAME : ANJI LAKSHMI
USN NO : 24BECS421
Enter Father's Age: 20
Enter Son's Age: 33
Son's age cannot be greater than or equal to father's age
Would you like to re-enter details (Y/n)

```

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.

Week - 08

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

```
class DisplayMessage1 extends Thread {  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("BMS college of engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println(e);  
        }  
    }  
  
    class DisplayMessage2 extends Thread {  
        public void run() {  
            try {  
                while (true) {  
                    System.out.println("CSE");  
                    Thread.sleep(2000);  
                }  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
        }  
    }  
  
    public class MultithreadingExample {  
        public static void main (String [] args) {  
            DisplayMessage1 thread1 = new DisplayMessage1();  
            DisplayMessage2 thread2 = new DisplayMessage2();  
        }  
    }  
}
```

thread1.start();

thread2.start();

Output :-

```
BMS college of engineering  
CS  
BMS college of engineering.  
CS  
CS  
BMS college of engineering  
CS
```

Code:

```
class DisplayMessage1 extends Thread {  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println(e);  
        }  
    }  
}  
  
class DisplayMessage2 extends Thread {  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println(e);  
        }  
    }  
}  
  
public class MultiThreadingExample {  
    public static void main(String[] args) {  
  
        DisplayMessage1 thread1 = new DisplayMessage1();  
        DisplayMessage2 thread2 = new DisplayMessage2();  
  
        thread1.start();  
        thread2.start();  
    }  
}
```

Output :

.

```
C:\Users\Admin\Sam\24BECs421>java MultiThreadingExample
BMS College of Engineering
CSE
BMS College of Engineering
CSE
BMS College of Engineering
BMS College of Engineering
```

Program 9

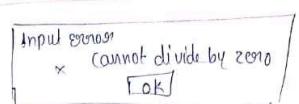
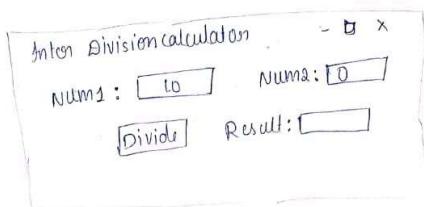
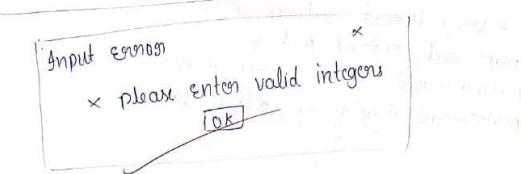
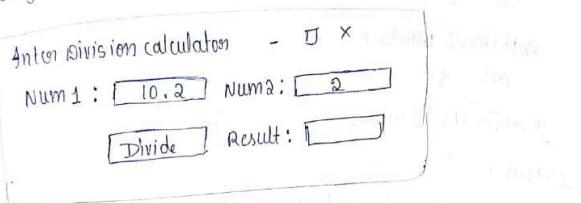
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException.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

```
block - 9  
import java.awt.*;  
import java.awt.event.*;  
import java.awt.event.ActionListener;  
import java.awt.event.ActionEvent;  
  
public class Divisioncalculator {  
    public static void main (String[] args) {  
        JFrame frame = new JFrame ("Integer Division calculator");  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.setSize (400,200);  
  
        JPanel panel = new JPanel();  
        panel.setLayout (new GridLayout ());  
  
        JLabel num1label = new JLabel ("Num1");  
        JTextField num1field = new JTextField (10);  
        JLabel num2label = new JLabel ("Num2");  
        JTextField num2field = new JTextField (10);  
        JLabel resultlabel = new JLabel ("Result:");  
        JTextField resultfield = new JTextField (10);  
        resultfield.setEditable (false);  
        JButton divideButton = new JButton ("Divide");  
  
        panel.add (num1label);  
        panel.add (num1field);  
        panel.add (num2label);  
        panel.add (num2field);  
        panel.add (divideButton);  
        panel.add (resultlabel);  
        panel.add (resultfield);  
    }  
}
```

```
frame.setVisible (true);  
divideButton.addActionListener (new ActionListener () {  
    @Override  
    public void actionPerformed (ActionEvent e) {  
        try {  
            String num1text = num1field.getText ();  
            String num2text = num2field.getText ();  
  
            int num1 = Integer.parseInt (num1text);  
            int num2 = Integer.parseInt (num2text);  
  
            if (num2 == 0) {  
                throw new ArithmeticException ("cannot divided by  
                0");  
            }  
            int result = num1 / num2;  
            resultfield.setText (String.valueOf (result));  
        } catch (NumberFormatException ex) {  
            JOptionPane.showMessageDialog (frame, "please enter valid  
            integers.", "Input Error", JOptionPane.ERROR_MESSAGE);  
        } catch (ArithmeticException ex) {  
            JOptionPane.showMessageDialog (frame, ex.getMessage(),  
                "Arithmetic Error", JOptionPane.ERROR_MESSAGE);  
        }  
    }  
});
```

Week - 4

- 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 displayed in a message dialog box.



Code :

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

public class DivisionCalculatorGUI {

    public static void main(String[] args) {

        JFrame frame = new JFrame("Integer Division Calculator");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 200);

        JPanel panel = new JPanel();
        panel.setLayout(new FlowLayout());

        JLabel num1Label = new JLabel("Num1:");
        JTextField num1Field = new JTextField(10);
        JLabel num2Label = new JLabel("Num2:");
        JTextField num2Field = new JTextField(10);
        JLabel resultLabel = new JLabel("Result:");
        JTextField resultField = new JTextField(10);
        resultField.setEditable(false); // Result field is read-only
        JButton divideButton = new JButton("Divide");

        panel.add(num1Label);
        panel.add(num1Field);
        panel.add(num2Label);
        panel.add(num2Field);
        panel.add(divideButton);
        panel.add(resultLabel);
        panel.add(resultField);

        frame.add(panel);

        frame.setVisible(true);

        divideButton.addActionListener(new ActionListener() {
            @Override

```

```

public void actionPerformed(ActionEvent e) {
    try {

        String num1Text = num1Field.getText();
        String num2Text = num2Field.getText();

        int num1 = Integer.parseInt(num1Text);
        int num2 = Integer.parseInt(num2Text);

        if (num2 == 0) {
            throw new ArithmeticException("Cannot divide by zero.");
        }
        int result = num1 / num2;

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

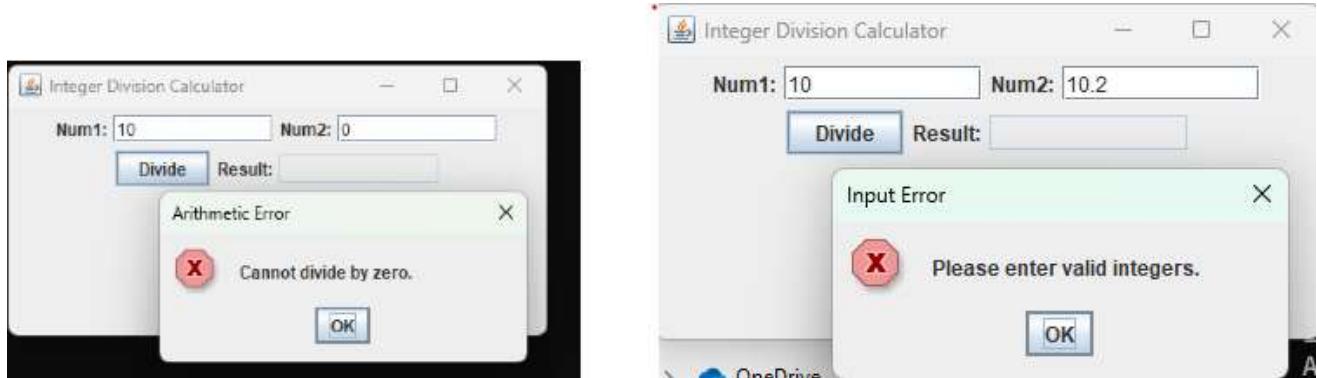
    } catch (NumberFormatException ex) {

        JOptionPane.showMessageDialog(frame, "Please enter valid integers.", "Input Error",
JOptionPane.ERROR_MESSAGE);
    } catch (ArithmeticException ex) {

        JOptionPane.showMessageDialog(frame, ex.getMessage(), "Arithmetic Error",
JOptionPane.ERROR_MESSAGE);
    }
}

```

Output :



Program 10

Demonstrate Inter process Communication and deadlock

Week :10

```
class Or {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while (!valueSet)
            try {
                System.out.println("In consumer waiting");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught in get()");
                Thread.currentThread().interrupt();
            }
        System.out.println("got : " + n);
        valueSet = false;
        System.out.println("In intimate producer");
        notify();
        return n;
    }
    synchronized void put(int n) {
        while (valueSet)
            try {
                System.out.println("In producer waiting");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught in put()");
                Thread.currentThread().interrupt();
            }
        this.n = n;
        valueSet = true;
        System.out.println("put : " + n);
        System.out.println("In intimate consumer");
        notify();
    }
}
```

```
class producer implements Runnable {
    Or o;
    private static final int max_items = 15;
    producer(Or o) {
        this.o = o;
        new Thread(this, "producer").start();
    }
    public void run() {
        int i = 0;
        while (i < max_items) {
            int n = o.get();
            System.out.println("consumed : " + n);
            i++;
        }
    }
}
```

```
class PCFixed {
    public static void main(String args[]) {
        Or o = new Or();
        new producer(o);
        new consumer(o);
        System.out.println("Press control -c to stop.");
    }
}
```

deadlock

```

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

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

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

    synchronized void last() {
        System.out.println("inside B.last()");
    }
}

```

class deadlock implements runnable {

```

A a = new A();
B b = new B();
deadlock() {
    Thread.currentThread().setName("main thread");
    Thread t = new Thread(this, "graceful thread");
    t.start();
    synchronized (a) {
        a.foo(b);
    }
    System.out.println("Back in main thread");
}

public void run() {
    synchronized (b) {
        b.bar(a);
    }
    System.out.println("Back in other thread");
}

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

```

Week 10

- (10) Demonstrate inter process communication & deadlock

Output:

put : 0

Intimate consumer

producer waiting

put : 0

Intimate producer

put : 1

Intimate consumer

Output:

Racing thread entered B . bar

main thread entered A . fo 0

Racing thread trying to call A . last()

main thread trying to call B . last()

Scen

121

124

124

Code

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

    synchronized int get() {
        while (!valueSet) {
            try {
                System.out.println("\nConsumer waiting");
                wait(); // Consumer waits if value is not set
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught in get()");
                Thread.currentThread().interrupt(); // Re-interrupt the thread
            }
        }
        System.out.println("Got: " + n);
        valueSet = false; // Mark the value as consumed
        System.out.println("\nIntimate Producer");
        notify(); // Notify producer to put new value
        return n;
    }

    synchronized void put(int n) {
        while (valueSet) {
            try {
                System.out.println("\nProducer waiting");
                wait(); // Producer waits if value has already been set
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught in put()");
                Thread.currentThread().interrupt(); // Re-interrupt the thread
            }
        }
        this.n = n;
        valueSet = true; // Mark the value as produced
        System.out.println("Put: " + n);
        System.out.println("\nIntimate Consumer");
        notify(); // Notify consumer that value is available
    }
}

class Producer implements Runnable {

    Q q;
    private static final int MAX_ITEMS = 15;

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

```

        new Thread(this, "Producer").start();
    }

    public void run() {
        int i = 0;
        while (i < MAX_ITEMS) { // Produce only up to MAX_ITEMS
            q.put(i++);
        }
    }
}

class Consumer implements Runnable {
    Q q;
    private static final int MAX_ITEMS = 15;

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

    public void run() {
        int i = 0;
        while (i < MAX_ITEMS) { // Consume only up to MAX_ITEMS
            int r = q.get();
            System.out.println("Consumed: " + r);
            i++;
        }
    }
}

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

```

Deadlock

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo {
    SwingDemo() {

```

```

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

jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

JLabel jlab = new JLabel("Enter the divider and divident:");

JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);

JButton button = new JButton("Calculate");

JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();

jfrm.add(err); // to display error bois
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

ActionListener l = new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        System.out.println("Action event from a text field");
    }
};

ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            int ans = a / b;
            alab.setText("\nA = " + a);
            blab.setText("\nB = " + b);
            anslab.setText("\nAns = " + ans);
        } catch (NumberFormatException e) {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter Only Integers!");
        }
    }
});

```

```
        } catch (ArithmaticException e) {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("B should be NON zero!");
        }
    });
    jfrm.setVisible(true);
}
public static void main(String args[]) {

    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SwingDemo();
        }
    });
}
```

output :

```
Producer waiting
Got: 11

Intimate Producer
Consumed: 11
Put: 12

Intimate Consumer

Producer waiting
Got: 12

Intimate Producer
Consumed: 12
Put: 13

Intimate Consumer

Producer waiting
Got: 13

Intimate Producer
Consumed: 13
Put: 14

Intimate Consumer
Got: 14

Intimate Producer
Consumed: 14
```

```
D:\24BECS400\week8>
```

```
D:\24BECS400\week8>javac PCFixed.java
```

```
D:\24BECS400\week8>java PCFixed
Press Control-C to stop.
Put: 0

Intimate Consumer

Producer waiting
Got: 0

Intimate Producer
Put: 1

Intimate Consumer

Producer waiting
Consumed: 0
Got: 1

Intimate Producer
Consumed: 1
Put: 2

Intimate Consumer

Producer waiting
Got: 2

Intimate Producer
Consumed: 2
Put: 3

Intimate Consumer

Producer waiting
Got: 3

Intimate Producer
Consumed: 3
Put: 4

Intimate Consumer

Producer waiting
Got: 4

Intimate Producer
Consumed: 4
Put: 5
```

```
D:\24BECS400\week8>javac Deadlock.java  
D:\24BECS400\week8>java Deadlock  
RacingThread entered B.bar  
MainThread entered A.foo  
RacingThread trying to call A.last()  
MainThread trying to call B.last()
```