VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
On

OBJECT ORIENTED JAVA PROGRAMMING (23CS3PCOOJ)

Submitted by

ZAIN MAHEDI (1WA23CS056)

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
September 2024-January 2025

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

This is to certify that the Lab work entitled **"OBJECT ORIENTED JAVA PROGRAMMING"** carried out by ZAIN MAHEDI **(1WA23CS056)**, who is bonafide student at **B. M. S. College of Engineering**. It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **OBJECT ORIENTED JAVA PROGRAMMING** Lab - **(23CS3PCOOJ)**work prescribed for the said degree.

Ms Ambuja K
Assistant Professor
Department of CSE
BMSCE, Bengaluru

Dr. Kavitha Sooda
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index Sheet

Course outcomes:

CO1	Apply the knowledge of Java concepts to find the solution for a given problem.
CO2	Analyse the given Java application for correctness/functionalities.
CO3	Develop Java programs / applications for a given requirement.
CO4	Conduct practical experiments for demonstrating features of Java.

Lab program 1

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions.

Program

```
import java.util.*;

class Quadratic {

   public static void main(String[] args) {

      int a, b, c;
      double discriminant;
      double r1, r2;
      Scanner sc = new Scanner(System.in);

      System.out.print("Enter the value for a: ");
      a = sc.nextInt();

      System.out.print("Enter the value for b: ");
      b = sc.nextInt();

      System.out.print("Enter the value for c: ");
      c = sc.nextInt();

      discriminant = b * b - (4 * a * c);

      if (discriminant == 0) {
         System.out.println("The discriminant is zero and has 2 equal roots");
         r1 = r2 = (double) -b / (2 * a);
         System.out.println("The roots are " + r1 + " & " + r2);
      } else if (discriminant > 0) {
         System.out.println("The discriminant is greater than zero and has 2 distinct roots");
         r1 = ((double) -b / (2 * a)) + Math.sqrt(discriminant);
         r2 = (double) -b / (2 * a) - Math.sqrt(discriminant);
         System.out.println("The roots are " + r1 + " & " + r2);
      } else {
         System.out.println("The discriminant is less than zero and has no real roots");
      }
      sc.close();
   }
}
```

Output

```
=============================================================================================

Enter the value for a: 1
Enter the value for b: 2
Enter the value for c: 1
The discriminant is zero and has 2 equal roots
The roots are -1.0 & -1.0
```

Lab program 2

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

Program

```
import java.util.Scanner;

class Student {
   String usn;
   String name;
   int[] sem1Credits;
   int[] sem1Marks;
   int[] sem2Credits;
   int[] sem2Marks;
   int numSubjectsSem1;
   int numSubjectsSem2;

   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.println("\nEntering the following details for sem 1:");
      System.out.print("Enter number of subjects for sem 1: ");
      numSubjectsSem1 = sc.nextInt();
      sem1Credits = new int[numSubjectsSem1];
      sem1Marks = new int[numSubjectsSem1];

      for (int i = 0; i < numSubjectsSem1; i++) {
         System.out.print("Enter credits for subject " + (i + 1) + " in sem 1: ");
         sem1Credits[i] = sc.nextInt();
         System.out.print("Enter marks for subject " + (i + 1) + " in sem 1: ");
         sem1Marks[i] = sc.nextInt();
      }

      System.out.println("\nEntering the following details for sem 2:");
      System.out.print("Enter number of subjects for sem 2: ");
      numSubjectsSem2 = sc.nextInt();
      sem2Credits = new int[numSubjectsSem2];
      sem2Marks = new int[numSubjectsSem2];
```

```
        for (int i = 0; i < numSubjectsSem2; i++) {
            System.out.print("Enter credits for subject " + (i + 1) + " in sem 2: ");
            sem2Credits[i] = sc.nextInt();
            System.out.print("Enter marks for subject " + (i + 1) + " in sem 2: ");
            sem2Marks[i] = sc.nextInt();
        }

        sc.close();
    }

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

        System.out.println("\nCredits and Marks for sem 1:");
        for (int i = 0; i < numSubjectsSem1; i++) {
            System.out.println("Subject " + (i + 1) + ": Credits = " + sem1Credits[i] + ", Marks =
" + sem1Marks[i]);
        }

        System.out.println("\nCredits and Marks for sem 2:");
        for (int i = 0; i < numSubjectsSem2; i++) {
            System.out.println("Subject " + (i + 1) + ": Credits = " + sem2Credits[i] + ", Marks =
" + sem2Marks[i]);
        }
    }

    void calculateSGPA() {
        System.out.println("\nSGPA Calculation:");

        double sgpaSem1 = calculateSemesterSGPA(sem1Credits, sem1Marks,
numSubjectsSem1);
        System.out.printf("SGPA for sem 1: %.2f\n", sgpaSem1);

        double sgpaSem2 = calculateSemesterSGPA(sem2Credits, sem2Marks,
numSubjectsSem2);
        System.out.printf("SGPA for sem 2: %.2f\n", sgpaSem2);

        double cgpa = (sgpaSem1 + sgpaSem2) / 2;
        System.out.printf("CGPA (Cumulative GPA): %.2f\n", cgpa);
    }

    double calculateSemesterSGPA(int[] credits, int[] marks, int numSubjects) {
        int totalCredits = 0;
        int weightedSum = 0;
```

```
        for (int i = 0; i < numSubjects; i++) {
            int gradePoint = calculateGradePoint(marks[i]);
            weightedSum += gradePoint * credits[i];
            totalCredits += credits[i];
        }
        return (double) weightedSum / totalCredits;
    }

    int calculateGradePoint(int marks) {
        if (marks >= 90)
            return 10; // O grade
        else if (marks >= 80)
            return 9; // A grade
        else if (marks >= 70)
            return 8; // B grade
        else if (marks >= 60)
            return 7; // C grade
        else if (marks >= 50)
            return 6; // D grade
        else if (marks >= 40)
            return 4; // P grade
        else
            return 0;
    }

    public static void main(String[] args) {
        Student student = new Student();
        student.acceptDetails();
        student.displayDetails();
        student.calculateSGPA();
    }
}
```

Output

```
=============================================================================================================


Enter USN: 1BM23CS001
Enter Name: Zain Mahedi

Entering the following details for sem 1:
Enter number of subjects for sem 1: 3
Enter credits for subject 1 in sem 1: 4
Enter marks for subject 1 in sem 1: 85
Enter credits for subject 2 in sem 1: 3
Enter marks for subject 2 in sem 1: 78
Enter credits for subject 3 in sem 1: 3
Enter marks for subject 3 in sem 1: 92

Entering the following details for sem 2:
Enter number of subjects for sem 2: 3
Enter credits for subject 1 in sem 2: 4
Enter marks for subject 1 in sem 2: 88
Enter credits for subject 2 in sem 2: 3
Enter marks for subject 2 in sem 2: 73
Enter credits for subject 3 in sem 2: 3
Enter marks for subject 3 in sem 2: 95

Student Details:
USN: 1BM23CS001
Name: Zain Mahedi

Credits and Marks for sem 1:
Subject 1: Credits = 4, Marks = 85
Subject 2: Credits = 3, Marks = 78
Subject 3: Credits = 3, Marks = 92

Credits and Marks for sem 2:
Subject 1: Credits = 4, Marks = 88
Subject 2: Credits = 3, Marks = 73
Subject 3: Credits = 3, Marks = 95

SGPA Calculation:
SGPA for sem 1: 8.70
SGPA for sem 2: 8.90
CGPA (Cumulative GPA): 8.80
```

Lab 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

Program

```
import java.util.Scanner;

class Book {
   String name;
   String author;
   double price;
   int num_pages;

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

   public String getName() {
      return name;
   }

   @Override
   public String toString() {
      return "Book Name " + name + " Author " + author + " Price " + price;
   }
}

class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String name, author, choice;
      double price;
      int num_pages, n;

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

```
        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + ":");
            System.out.print("Name : ");
            name = sc.nextLine();
            System.out.print("Author : ");
            author = sc.nextLine();
            System.out.print("Price : ");
            price = sc.nextDouble();
            System.out.print("Number of pages : ");
            num_pages = sc.nextInt();
            sc.nextLine();

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

        System.out.print("Enter name of book to search :");
        choice = sc.nextLine();

        boolean found = false;
        for (int i = 0; i < n; i++) {
            if (books[i].getName().equalsIgnoreCase(choice)) {
                System.out.println(books[i].toString());
                found = true;
                break;
            }
        }

        if (!found) {
            System.out.println("Book not found!");
        }

        sc.close();

    }
}
```

Output

```
=============================================================================================================
Enter the number of books : 2

Book 1:
Name : Java Programming
Author : John Doe
Price : 500
Number of pages : 350

Book 2:
Name : Python Programming
Author : Jane Smith
Price : 450
Number of pages : 300

Enter name of book to search : Java Programming
Book Name Java Programming Author John Doe Price 500.0

Enter the number of books : 2

Book 1:
Name : Java Programming
Author : John Doe
Price : 500
Number of pages : 350

Book 2:
Name : Python Programming
Author : Jane Smith
Price : 450
Number of pages : 300

Enter name of book to search : C Programming
Book not found!
```

Lab program 4

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

Program

```
abstract class Shape {
   int d1;
   int d2;

   Shape(int d1, int d2) {
      this.d1 = d1;
      this.d2 = d2;
   }

   abstract void printArea();
}

class Rectangle extends Shape {

   Rectangle(int l, int b) {
      super(l, b);
   }

   void printArea() {
      System.out.println("Rectangle Shape");
      System.out.println("The area is : " + d1 * d2);
   }
}

class Triangle extends Shape {

   Triangle(int b, int h) {
      super(b, h);
   }

   void printArea() {
      System.out.println("Triangle Shape");
      System.out.println("The area is : " + 0.5 * d1 * d2);
   }
}
```

```
class Circle extends Shape {

    Circle(int r) {
        super(r, 0);
    }

    void printArea() {
        System.out.println("Circle Shape");
        System.out.println("The area is : " + (Math.PI * d1 * d1));
    }
}

class Main {

    public static void main(String[] args) {
        Shape shape;

        shape = new Rectangle(5, 2);
        shape.printArea();

        shape = new Triangle(5, 2);
        shape.printArea();

        shape = new Circle(5);
        shape.printArea();
    }
}
```

Output

```
=====================================================================================================
Rectangle Shape
The area is : 10

Triangle Shape
The area is : 5.0

Circle Shape
The area is : 78.53981633974483
```

Lab program 5

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

Program

```
import java.util.Scanner;

class Bank {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String name, accountNumber, accountType;
      int choice;
      double amount;

      System.out.println("Welcome to the Bank\n");

      System.out.print("Enter the name of the account holder: ");
      name = sc.nextLine();

      System.out.print("Enter the account number: ");
      accountNumber = sc.nextLine();

      System.out.println("Choose your account type:");
      System.out.println("1. Savings Account\t2. Current Account");
      System.out.print("Enter your choice: ");
      choice = sc.nextInt();
```

```
        System.out.println("Thank you for registering!");

        Account account;
        if (choice == 1) {
            account = new SavingsAccount();
            accountType = "Savings Account";
        } else {
            account = new CurrentAccount();
            accountType = "Current Account";
        }

        account.customerName = name;
        account.accountNumber = accountNumber;
        account.accountType = accountType;

        while (true) {
            System.out.println("\n1. Deposit\n2. Withdraw\n3. Get Balance\n4. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter the amount to deposit: ");
                    amount = sc.nextDouble();
                    account.deposit(amount);
                    break;
                case 2:
                    System.out.print("Enter the amount to withdraw: ");
                    amount = sc.nextDouble();
                    account.withdraw(amount);
                    break;
                case 3:
                    account.getBalance();
                    break;
                case 4:
                    System.out.println("Exiting...");
                    sc.close();
                    System.exit(0);
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }
    }
}

abstract class Account {
```

```
    String customerName;
    String accountNumber;
    String accountType;
    Double balance = 0.0;

    abstract void deposit(double amount);

    abstract void withdraw(double amount);

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

class CurrentAccount extends Account {
    double minBalance = 8500.0;
    double penalty = 2000.0;

    void deposit(double amount) {
        if (balance == 0) {
            if (amount < minBalance) {
                System.out.println("The minimum deposit is " + minBalance);
                System.out.println("Please deposit a higher amount");
                return;
            }
        }
        balance += amount;
        System.out.println("Deposited " + amount + " to " + customerName + ", Current
Balance: " + balance);
    }

    void withdraw(double amount) {
        if (amount > minBalance) {
            balance -= penalty;
            System.out.println("You have been penalized for not maintaining the minimum
balance");
        } else {
            balance -= amount;
            System.out.println("Withdrawn " + amount + " from " + customerName);
        }
    }
}

class SavingsAccount extends Account {
    final double CI = 0.05;
```

```
    void deposit(double amount) {
        balance += amount + (amount * CI);
        System.out.println("Deposited " + amount + " to " + customerName + ", Current
Balance: " + balance);
        System.out.println("Compound Interest earned = " + (amount * CI));
    }

    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance");
        } else {
            balance -= amount;
            System.out.println("Withdrawn " + amount + " from " + customerName);
        }
    }
}
```

Output

```
================================================================================================================================

Welcome to the Bank

Enter the name of the account holder: Zain Mahedi
Enter the account number: 12345678
Choose your account type:
1. Savings Account  2. Current Account
Enter your choice: 2
Thank you for registering!

1. Deposit
2. Withdraw
3. Get Balance
4. Exit
Enter your choice: 1
Enter the amount to deposit: 5000
The minimum deposit is 8500.0
Please deposit a higher amount

1. Deposit
2. Withdraw
3. Get Balance
4. Exit
Enter your choice: 1
Enter the amount to deposit: 10000
Deposited 10000.0 to Zain Mahedi, Current Balance: 10000.0

1. Deposit
2. Withdraw
3. Get Balance
4. Exit
Enter your choice: 2
Enter the amount to withdraw: 9000
You have been penalized for not maintaining the minimum balance

1. Deposit
2. Withdraw
3. Get Balance
4. Exit
Enter your choice: 3
Current Balance: 8000.0

1. Deposit
2. Withdraw
3. Get Balance
4. Exit
Enter your choice: 2
Enter the amount to withdraw: 8500
You have been penalized for not maintaining the minimum balance

1. Deposit
2. Withdraw
3. Get Balance
4. Exit
Enter your choice: 2
Enter the amount to withdraw: 1000
Withdrawn 1000.0 from Zain Mahedi

1. Deposit
2. Withdraw
3. Get Balance
4. Exit
Enter your choice: 3
Current Balance: 7000.0

1. Deposit
2. Withdraw
3. Get Balance
4. Exit
Enter your choice: 4
Exiting...
```

Lab program 6

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals have 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.

Program

Main.java

```
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();

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Student " + (i + 1) + ":");
            scanner.nextLine(); // Consume newline

            System.out.print("USN: ");
            String usn = scanner.nextLine();

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

            System.out.print("Semester: ");
            int sem = scanner.nextInt();

            System.out.println("Enter internal marks for 5 courses (out of 50): ");
            int[] internalMarks = new int[5];
            for (int j = 0; j < 5; j++) {
                System.out.print("Course " + (j + 1) + " internal marks: ");
                internalMarks[j] = scanner.nextInt();
            }
```

```
        @SuppressWarnings("unused")
        Internals internals = new Internals(internalMarks);

        System.out.println("Enter SEE marks for 5 courses (out of 100): ");
        int[] seeMarks = new int[5];
        for (int j = 0; j < 5; j++) {
            System.out.print("Course " + (j + 1) + " SEE marks: ");
            seeMarks[j] = scanner.nextInt();
        }

        External external = new External(usn, name, sem, seeMarks);

        System.out.println("\nFinal Marks for " + external.getName() + " (" +
external.getUsn() + "):");
        System.out.println("Semester: " + external.getSem());
        System.out.println("Internal Marks:");
        for (int j = 0; j < 5; j++) {
            double finalMarks = internalMarks[j] + (external.getSeeMarks()[j] / 2.0);
            System.out.println("Course " + (j + 1) + ": " + finalMarks);
        }
    }
    scanner.close();
  }
}
```

see/External.java

```
package see;
import cie.Student;

public class External extends Student {
  private int[] seeMarks = new int[5];

  public External(String usn, String name, int sem, int[] seeMarks) {
    super(usn, name, sem);
    for (int i = 0; i < 5; i++) {
      this.seeMarks[i] = seeMarks[i];
    }
  }

  public int[] getSeeMarks() {
    return seeMarks;
  }
}
```

cie/Internals.java

```
package cie;
```

```
public class Internals {
    protected int[] internalMarks = new int[5];

    public Internals(int[] marks) {
        for (int i = 0; i < 5; i++) {
            internalMarks[i] = marks[i];
        }
    }

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

cie/Student.java

```
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;
    }
}
```

Output

```
=============================================================================================================

Enter the number of students: 1

Enter details for Student 1:
USN: 1WA23CS056
Name: Zain Mahedi
Semester: 5

Enter internal marks for 5 courses (out of 50):
Course 1 internal marks: 40
Course 2 internal marks: 35
Course 3 internal marks: 30
Course 4 internal marks: 45
Course 5 internal marks: 50

Enter SEE marks for 5 courses (out of 100):
Course 1 SEE marks: 80
Course 2 SEE marks: 70
Course 3 SEE marks: 60
Course 4 SEE marks: 90
Course 5 SEE marks: 100

Final Marks for Zain Mahedi (1WA23CS056):
Semester: 5
Internal Marks:
Course 1: 80.0
Course 2: 70.0
Course 3: 60.0
Course 4: 90.0
Course 5: 100.0
```

Lab 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 Wrong Age( ) when the input age=father's age.

Program

```
import java.util.Scanner;

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

class Father {
    public Father(int ageFather) throws WrongAge {
        if (ageFather < 0) {
            throw new WrongAge("Father's age should be greater than or equal to 0.");
        }
    }
}

class Son extends Father {
    public Son(int ageFather, int ageSon) throws WrongAge {
        super(ageFather);
        if (ageSon >= ageFather) {
            throw new WrongAge("Son's age should be less than the father's age.");
        }
    }
}

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

        System.out.print("Enter father's age: ");
        int ageFather = sc.nextInt();

        System.out.print("Enter son's age: ");
        int ageSon = sc.nextInt();
```

```
try {
            new Son(ageFather, ageSon);
            System.out.println("Father's and Son's ages are valid.");
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        }

        sc.close();
    }
}
```

Output

```
================================================================================================================

Enter father's age: 40
Enter son's age: 45
Exception: Son's age should be less than the father's age.
```

Lab 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.

Program

```
class MyThread extends Thread {
   final String message;
   final int interval;
   MyThread(String message, int interval) {
      this.message = message;
      this.interval = interval;
   }
   @Override
   public void run() {
      while (true) {
         try {
            System.out.println(message);
            Thread.sleep(interval * 1000);
         } catch (InterruptedException e) {
            System.out.println(e.getMessage());
         }
      }
   }
}
class Main {
   public static void main(String[] args) {
      MyThread thread1 = new MyThread("BMS College of Engineering",
            10);
      MyThread thread2 = new MyThread("CSE", 2);
      thread1.start();
      thread2.start();
   }
}
```

Output

```
================================================================================================================

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

Lab program 9

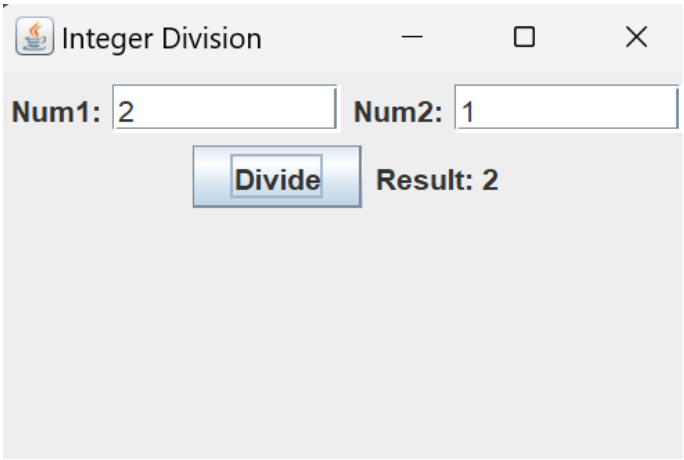
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number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

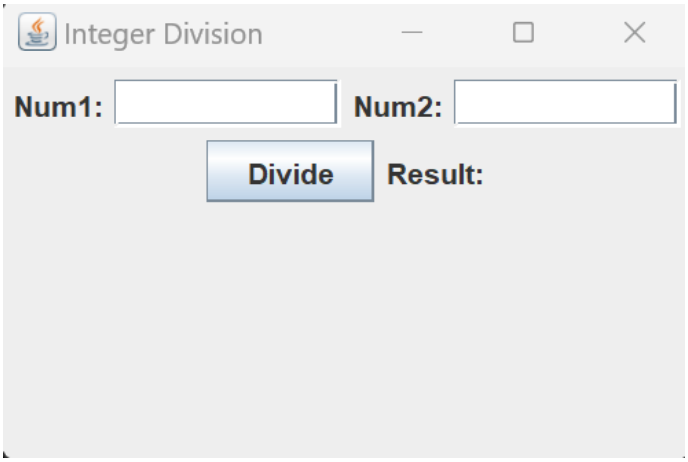
Program

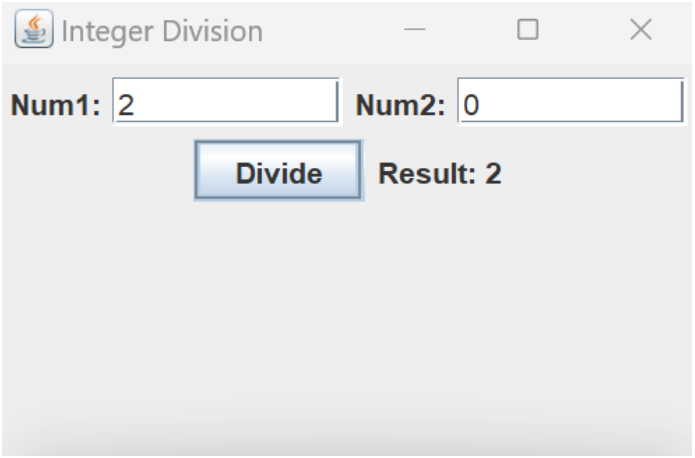
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

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

      JLabel label1 = new JLabel("Num1:");
      JTextField num1Field = new JTextField(10);
      JLabel label2 = new JLabel("Num2:");
      JTextField num2Field = new JTextField(10);
      JButton divideButton = new JButton("Divide");
      JLabel resultLabel = new JLabel("Result:");

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

      divideButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            try {
               int num1 = Integer.parseInt(num1Field.getText());
               int num2 = Integer.parseInt(num2Field.getText());
               if (num2 == 0) {
                  throw new ArithmeticException("Cannot divide by zero");
```

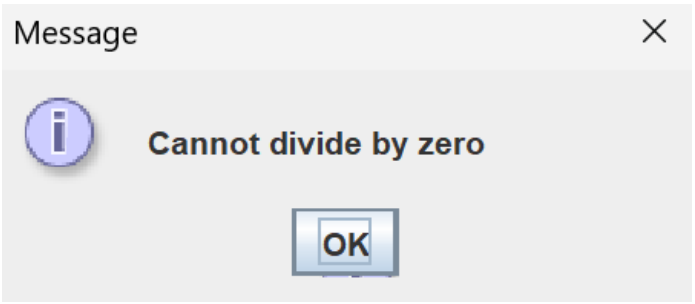
```
                }

                int result = num1 / num2;
                resultLabel.setText("Result: " + result);
            } catch (NumberFormatException ex) {
                JOptionPane.showMessageDialog(frame, "Invalid input! Please enter integers
only.");
            } catch (ArithmeticException ex) {
                JOptionPane.showMessageDialog(frame, ex.getMessage());
            }
        }
    });

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

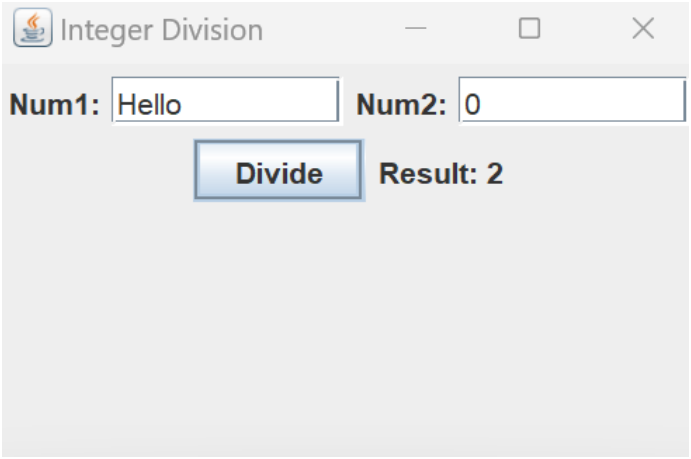
Output






Integer Division
Num1: 2
Num2: 0
Divide
Result: 2


Message
Cannot divide by zero
OK


Integer Division
Num1: Hello
Num2: 0
Divide
Result: 2


Message
Invalid input! Please enter integers only.
OK

Lab program 10

Demonstrate Inter process Communication and deadlock

Program

Inter Process Communication

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

   synchronized int get() {
      while (!valueSet)
         try {
            wait();
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
         }
      System.out.println("Got: " + n);
      valueSet = false;
      notify();
      return n;
   }

   synchronized void put(int n) {
      while (valueSet)
         try {
            wait();
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
         }
      this.n = n;
      valueSet = true;
      System.out.println("Put: " + n);
      notify();
   }
}

class Producer implements Runnable {
   Q q;

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

   public void run() {
```

```
        int i = 0;
        while (true) {
            q.put(i++);
        }
    }
}

class Consumer implements Runnable {
    Q q;

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

    public void run() {
        while (true) {
            q.get();
        }
    }
}

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

Output

```
================================================================================================

Press Control-C to stop.
Put: 0
Got: 0
Put: 1
Got: 1
Put: 2
Got: 2
Put: 3
Got: 3
Put: 4
Got: 4
```

Deadlock

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

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

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

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

class Deadlock implements Runnable {
   AA a = new AA();
   BB b = new BB();

   Deadlock() {
      Thread.currentThread().setName("MainThread");
      Thread t = new Thread(this, "RacingThread");
      t.start();
```

```
        a.foo(b); // Get lock on AA in the main thread.
        System.out.println("Back in main thread");
    }

    public void run() {
        b.bar(a); // Get lock on BB in the other thread.
        System.out.println("Back in other thread");
    }

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

Output

```
================================================================================================================

MainThread entered AA.foo
RacingThread entered BB.bar
MainThread trying to call BB.last()
RacingThread trying to call AA.last()
```