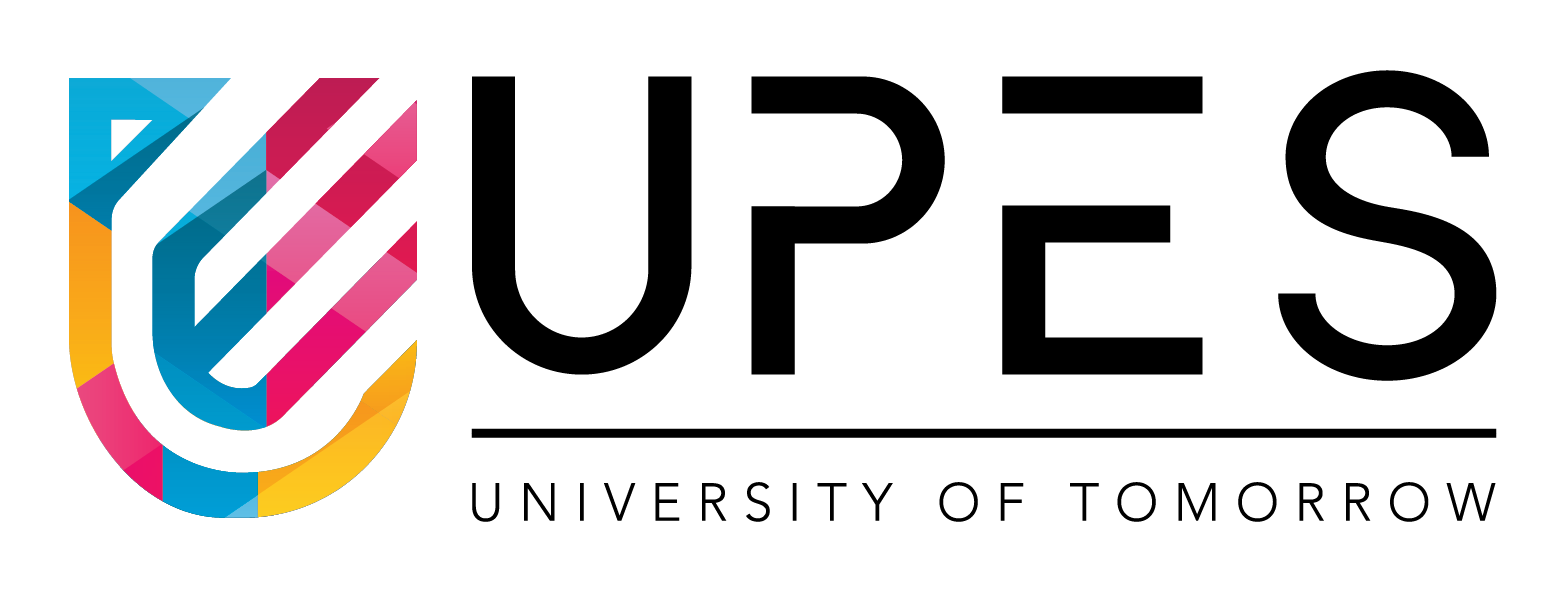
**OBJECT ORIENTED PROGRAMMING LAB**



Name – BHAVYA TALWAR

SAP ID – 500121992

Course – BTech CSE AIML

BATCH-03

Submitted To– Mr. Saurabh Jain

**Experiment - 08**

1. Create an exception class, which throws an exception if the operand is non-numeric in calculating modules. (Use command line arguments).

**Code:**

class NonNumericException extends Exception {

    public NonNumericException(String message) {

*super*(message);

    }

}

public class ques1 {

    static double Numeric(String arg) throws NonNumericException {

        try {

            return Double.parseDouble(arg);

        } catch (NumberFormatException e) {

            throw new NonNumericException("Invalid numeric operand: " + arg);

        }

    }

    public static void main(String[] args) {

        try {

            double a = Numeric(args[0]);

            double b = Numeric(args[1]);

            double sum = a + b;

            System.out.println("Sum: " + sum);

        } catch (NonNumericException e) {

            System.out.println("Error: " + e.getMessage());

        }

        finally{

            System.out.println("Bhavya Talwar");

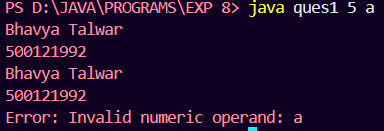
            System.out.println("500121992");

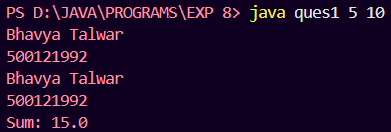
        }

    }

}

**Output:**





1. Write a Java program to throw an exception for employee details.

* If an employee’s name is a number, a name exception must be thrown.
* If an employee’s age exceeds 50, an age exception must be thrown.

**Code:**

import java.util.Scanner;

public class ques2 {

    String name;

    int age;

    ques2(String name, int age) throws Exception {

        if (!name.matches("[a-zA-Z]+")) {

            throw new Exception("Invalid Name: Your name contains numbers.");

        }

        if (age > 50) {

            throw new Exception("Invalid Age: " + age + " exceeds the limit of 50.");

        }

*this*.name = name;

*this*.age = age;

    }

    public void display() {

        System.out.println("Employee Name: " + name);

        System.out.println("Employee Age: " + age);

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

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

        String name = scanner.nextLine();

        System.out.print("Enter Employee Age: ");

        int age = scanner.nextInt();

        try {

            ques2 emp = new ques2(name, age);

            emp.display();

        } catch (Exception e) {

            System.out.println("Error: " + e.getMessage());

        } finally {

            System.out.println("Bhavya Talwar");

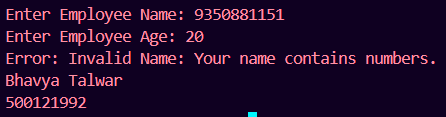
            System.out.println("500121992");

        }

    }

}

**Output:**



1. Write a Java program that takes two integers as input from the user and performs division. Handle the ArithmeticException that occurs if the denominator is zero. Use a try-catch block to catch the exception and display an appropriate error message. Additionally, use a finally block to print "Operation completed" regardless of whether an exception occurs or not.

**Code:**

import java.util.Scanner;

class ques3 {

    public static void main(String[] args) {

        try {

            Scanner sc = new Scanner(System.in);

            System.out.print("Enter 1st number: ");

            int a = sc.nextInt();

            System.out.print("Enter 2nd number: ");

            int b = sc.nextInt();

            double div = a / b;

            System.out.println("The quotient is: " + div);

        } catch (ArithmeticException e) {

            System.out.println("Error: Division by zero is not allowed.");

        } finally {

            System.out.println("Operation completed.");

            System.out.println("Bhavya Talwar");

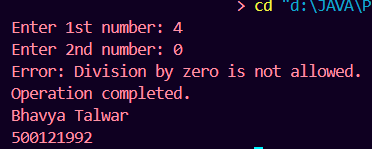
            System.out.println("500121992");

        }

    }

}

**Output:**

****

1. Write a Java program that creates an array of 5 integers and asks the user to enter an index to access the array element. Handle the ArrayIndexOutOfBoundsException if the user enters an invalid index. Use a try-catch block to catch the exception and display an appropriate error message. Use the finally block to print "Array access attempted.

**Code:**

import java.util.Scanner;

public class ques4 {

    public static void main(String[] args) {

        int[] a = {10, 20, 30, 40, 50};

        Scanner sc = new Scanner(System.in);

        try {

            System.out.print("Enter an index to access the array element: ");

            int index = sc.nextInt();

            System.out.println("Element at index " + index + ": " + a[index]);

        } catch (ArrayIndexOutOfBoundsException e) {

            System.out.println("Error: Invalid index! Please enter a value between 0 and 4.");

        } finally {

            System.out.println("Array access attempted.");

            System.out.println("Bhavya Talwar");

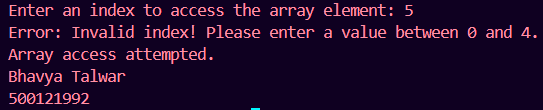
            System.out.println("500121992");

        }

    }

}

**Output:**

****