**Experiment: 7**

PART B

(PART B: TO BE Submitted BY STUDENTS)

**Aim : Exception handling, Difference between exception and error, try, catch, finally, throw, throws, finally**.

**Task:**

1. Write a program that takes user input for a number. Ensure that the input is a valid integer. If the input is not a valid integer (e.g., letters or special characters), catch the number format exception and display an appropriate error message. Use the finally block to print a message indicating that the input process is complete.

2. Write a program that prompts the user to input two numbers and then divides the first number by the second number. Handle the case where the second number is zero by catching the arithmetic exception and displaying a message such as "Cannot divide by zero." Ensure that the program handles valid input, even when a zero divisor is entered.

3. Write a program that asks the user to input an index to access an array of size 5. If the user enters an invalid index (i.e., an index less than 0 or greater than the array size), catch the array index out of bounds exception and print an error message. Allow the user to retry entering a valid index.

4. Write a program that takes two inputs from the user: a number and an index. The program should perform the following checks:

1. If the input number is not an integer, throw a number format exception.
2. If the index is out of bounds for an array of size 5, throw an array index out of bounds exception.
3. If there is an attempt to divide by zero when performing an operation, throw an arithmetic exception.
4. Write a banking program where the user can deposit and withdraw money from their account. Create a custom exception InsufficientBalanceException to handle cases where the withdrawal amount is greater than the available balance. The program should also ensure that the deposit amount is not negative.
5. Write a program to handle exception of your choice using throws.

**PART B**

(PART B: TO BE COMPLETED BY STUDENTS)

Students must execute all the tasks in Experiment-7 and copy paste the code, along with the snapshot of the output in Part-B. Upload and Submit the Part-B in soft copy on the portal. The filename should be **OOPJ\_batch\_rollno\_experimentno Example: OOPJ\_A1\_E203\_P7**

|  |  |
| --- | --- |
| **Roll No.: C126** | **Name: Rushabh Abhay Shah** |
| **Prog/Yr/Sem: BTI/4/8** | **Batch: D1** |
| **Date of Experiment: 15-03-2025** | **Date of Submission: 15-03-2025** |

Q1

import java.util.InputMismatchException;

import java.util.Scanner;

public class q1 {

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

        Scanner scanner = new Scanner(System.in);

        try {

            System.out.print("Enter a valid integer: ");

            int number = scanner.nextInt();

            System.out.println("You entered: " + number);

        } catch (InputMismatchException *e*) {

            System.out.println("Error: Input is not a valid integer.");

        } finally {

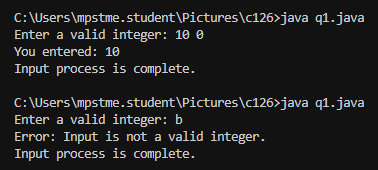
            System.out.println("Input process is complete.");

            scanner.close();

        }

    }

}



Q2

import java.util.InputMismatchException;

import java.util.Scanner;

public class q2 {

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

        Scanner scanner = new Scanner(System.in);

        try {

            System.out.print("Enter the first number: ");

            double num1 = scanner.nextDouble();

            System.out.print("Enter the second number: ");

            double num2 = scanner.nextDouble();

            double result = num1 / num2;

            System.out.println("Result: " + result);

        } catch (ArithmeticException *e*) {

            System.out.println("Cannot divide by zero.");

        } catch (InputMismatchException *e*) {

            System.out.println("Error: Invalid input. Please enter numbers.");

        } finally {

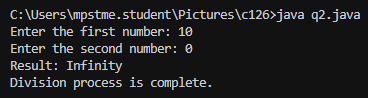
            System.out.println("Division process is complete.");

            scanner.close();

        }

    }

}



Q3

import java.util.InputMismatchException;

import java.util.Scanner;

public class q3 {

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

        Scanner scanner = new Scanner(System.in);

        int[] array = {1, 2, 3, 4, 5};

        boolean validIndex = false;

        while (!validIndex) {

            try {

                System.out.print("Enter an index (0-4): ");

                int index = scanner.nextInt();

                if (index < 0 || index >= array.length) {

                    throw new ArrayIndexOutOfBoundsException();

                }

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

                validIndex = true;

            } catch (ArrayIndexOutOfBoundsException *e*) {

                System.out.println("Error: Index out of bounds. Please enter a number between 0 and 4.");

            } catch (InputMismatchException *e*) {

                System.out.println("Error: Invalid input. Please enter an integer.");

                scanner.next();

            }

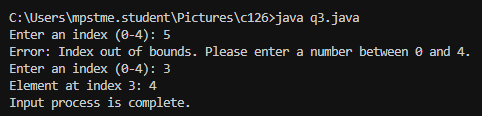
        }

        System.out.println("Input process is complete.");

        scanner.close();

    }

}



Q4

import java.util.InputMismatchException;

import java.util.Scanner;

public class q4 {

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

        Scanner scanner = new Scanner(System.in);

        try {

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

            int number = scanner.nextInt();

            System.out.print("Enter an index (0-4): ");

            int index = scanner.nextInt();

            if (index < 0 || index >= 5) {

                throw new ArrayIndexOutOfBoundsException();

            }

            if (number == 0) {

                throw new ArithmeticException();

            }

            double result = 10 / number;

            System.out.println("Result: " + result);

            System.out.println("Index is valid.");

        } catch (InputMismatchException *e*) {

            System.out.println("Error: Input is not a valid integer.");

        } catch (ArrayIndexOutOfBoundsException *e*) {

            System.out.println("Error: Index out of bounds.");

        } catch (ArithmeticException *e*) {

            System.out.println("Error: Attempt to divide by zero.");

        } finally {

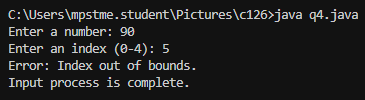
            System.out.println("Input process is complete.");

            scanner.close();

        }

    }

}



Q5

class InsufficientBalanceException extends Exception {

    public InsufficientBalanceException(String *message*) {

        super(message);

    }

}

class BankAccount {

    private double balance;

    public BankAccount(double *initialBalance*) {

        if (initialBalance < 0) {

            throw new IllegalArgumentException("Initial balance cannot be negative");

        }

        this.balance = initialBalance;

    }

    public void deposit(double *amount*) {

        if (amount < 0) {

            System.out.println("Deposit amount cannot be negative.");

            return;

        }

        balance += amount;

        System.out.println("Deposited: " + amount);

        displayBalance();

    }

    public void withdraw(double *amount*) throws InsufficientBalanceException {

        if (amount > balance) {

            throw new InsufficientBalanceException("Insufficient balance for withdrawal");

        }

        balance -= amount;

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

        displayBalance();

    }

    public void displayBalance() {

        System.out.println("Current Balance: " + balance);

    }

}

public class q5 {

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

        BankAccount account = new BankAccount(1000.0);

        account.deposit(500.0);

        account.deposit(-200.0);

        try {

            account.withdraw(200.0);

            account.withdraw(1500.0);

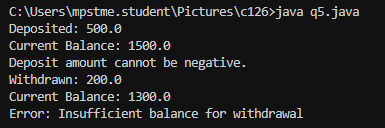
        } catch (InsufficientBalanceException *e*) {

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

        }

    }

}



Q6

import java.util.Scanner;

public class q6 {

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

        Scanner scanner = new Scanner(System.in);

        try {

            System.out.print("Enter a number to divide by: ");

            int divisor = scanner.nextInt();

            divide(divisor);

        } catch (ArithmeticException *e*) {

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

        }

        try {

            System.out.print("Enter an index for an array of size 5: ");

            int index = scanner.nextInt();

            accessArray(index);

        } catch (ArrayIndexOutOfBoundsException *e*) {

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

        }

    }

    public static void divide(int *divisor*) throws ArithmeticException {

        if (divisor == 0) {

            throw new ArithmeticException("Cannot divide by zero.");

        }

        System.out.println("Division successful.");

    }

    public static void accessArray(int *index*) throws ArrayIndexOutOfBoundsException {

        int[] array = {1, 2, 3, 4, 5};

        if (index < 0 || index >= array.length) {

            throw new ArrayIndexOutOfBoundsException("Index out of bounds.");

        }

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

    }

}

