Exception Handling

Q1. Electricity Bill Calculation with Exception Handling

Design a Java program to calculate the electricity bill for a customer, including exception handling for invalid input values. Implement a class named ElectricityBill with the following specifications:

Class: ElectricityBill

Instance Variables

● customerName (String): Name of the customer

● unitsConsumed (double): Number of electricity units consumed

● billAmount (double): The calculated bill amount

Constructor

● A parameterized constructor to initialize the customerName and unitsConsumed fields.

● Throw an IllegalArgumentException if unitsConsumed is negative.

Method

● void calculateBillAmount(): This method calculates the electricity bill based on the following rules:

○ First 100 units: Rs. 5 per unit

○ Next 200 units (101–300): Rs. 7 per unit

○ Above 300 units: Rs. 10 per unit

Main Program

In the main()

method:

1. Prompt the user to enter the customer's name and units consumed.

2. Use try-catch blocks to handle the following scenarios:

○ Catch InputMismatchException if the user enters non-numeric data for units.

○ Catch IllegalArgumentException if a negative value is entered for units.

3. If the input is valid, create an object of the ElectricityBill class, compute the bill using calculateBillAmount(), and print the customer's name, units consumed, and the total bill amount.

```
import java.util.InputMismatchException;

import java.util.Scanner;


class ElectricityBill {
```

```
private String customerName;
private double unitsConsumed;
private double billAmount;

// Constructor to initialize customer name and units consumed
ElectricityBill(String customerName, double unitsConsumed) {
    if (unitsConsumed < 0) {
        throw new IllegalArgumentException("Units consumed cannot be negative.");
    }
    this.customerName = customerName;
    this.unitsConsumed = unitsConsumed;
    this.billAmount = 0; // Initialize bill amount
}

// Method to calculate the electricity bill
public void calculateBillAmount() {
    if (unitsConsumed <= 100) {
        billAmount = unitsConsumed * 5;
    } else if (unitsConsumed <= 300) {
        billAmount = (100 * 5) + ((unitsConsumed - 100) * 7);
    } else {
        billAmount = (100 * 5) + (200 * 7) + ((unitsConsumed - 300) * 10);
    }
}

// Method to display the bill details
public void displayBill() {
    System.out.println("\nElectricity Bill Details:");
    System.out.println("Customer Name: " + customerName);
    System.out.println("Units Consumed: " + unitsConsumed);
    System.out.println("Total Bill Amount: Rs. " + billAmount);
```

```
    }
}

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

        try {
            // Taking user input
            System.out.print("Enter Customer Name: ");
            String name = scanner.nextLine();

            System.out.print("Enter Units Consumed: ");
            double units = scanner.nextDouble();

            // Creating object of ElectricityBill class
            ElectricityBill bill = new ElectricityBill(name, units);

            // Calculating bill amount
            bill.calculateBillAmount();

            // Displaying bill details
            bill.displayBill();

        } catch (InputMismatchException e) {
            System.out.println("Error: Please enter a valid numeric value for units consumed.");
        } catch (IllegalArgumentException e) {
            System.out.println("Error: " + e.getMessage());
        } finally {
        }
```

```
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 5>javac ElectricityBillDemo.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 5>java ElectricityBillDemo
Enter Customer Name: aman
Enter Units Consumed: 586

Electricity Bill Details:
Customer Name: aman
Units Consumed: 586.0
Total Bill Amount: Rs. 4760.0
```

Q2. Student Marks and Grade Calculation with Exception Handling

Design a Java program to calculate the total marks, average, and grade of a student, with proper exception handling for invalid inputs. Implement a class named Student with the following specifications:

Class: Student

Instance Variables

● name (String): Name of the student

● rollNo (int): Roll number of the student

● marks (double array of size 5): Marks obtained in 5 subjects

● average (double): Average marks

● grade (char): Grade based on average

Constructor

● A parameterized constructor to initialize the name, rollNo, and marks.

● Throw an IllegalArgumentException if any mark is negative or greater than 100.

Methods

● void calculateAverage(): Computes the average of marks.

● void calculateGrade(): Assigns grade based on the average as per the following criteria:

○ A: average ≥ 90 ○ B: 80 ≤ average < 90

○ C: 70 ≤ average < 80

○ D: 60 ≤ average < 70

○ F: average < 60

● void displayStudentInfo(): Displays the student's name, roll number, marks, average, and grade.

Main Program

In the main() method:

1. Prompt the user to input student details and marks for 5 subjects.

2. Use a try-catch block to handle the following:

○ InputMismatchException for non-numeric input

○ IllegalArgumentException for invalid mark entries (e.g., < 0 or > 100)

3. Create a Student object, calculate average and grade, and display the full information.

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

class Student {
    private String name;
    private int rollNo;
    private double[] marks = new double[5]; // Marks for 5 subjects
    private double average;
    private char grade;

    // Constructor to initialize student details
    Student(String name, int rollNo, double[] marks) {
        this.name = name;
        this.rollNo = rollNo;

        // Validate marks (must be between 0 and 100)
        for (int i = 0; i < marks.length; i++) {
            if (marks[i] < 0 || marks[i] > 100) {
                throw new IllegalArgumentException("Marks should be between 0 and 100.");
            }
        }
        this.marks = marks;
        this.average = 0;
        this.grade = 'F'; // Default grade
```

```
}

// Method to calculate average marks
public void calculateAverage() {
    double sum = 0;
    for (double mark : marks) {
        sum += mark;
    }
    average = sum / marks.length;
}

// Method to calculate grade based on average
public void calculateGrade() {
    if (average >= 90) {
        grade = 'A';
    } else if (average >= 80) {
        grade = 'B';
    } else if (average >= 70) {
        grade = 'C';
    } else if (average >= 60) {
        grade = 'D';
    } else {
        grade = 'F';
    }
}

// Method to display student information
public void displayStudentInfo() {
    System.out.println("\nStudent Details:");
    System.out.println("Name: " + name);
    System.out.println("Roll Number: " + rollNo);
```

```
        System.out.print("Marks: ");
        for (double mark : marks) {
            System.out.print(mark + " ");
        }
        System.out.println("\nAverage Marks: " + average);
        System.out.println("Grade: " + grade);
    }
}

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

        try {
            // Taking user input
            System.out.print("Enter Student Name: ");
            String name = scanner.nextLine();

            System.out.print("Enter Roll Number: ");
            int rollNo = scanner.nextInt();

            double[] marks = new double[5];
            System.out.println("Enter marks for 5 subjects (out of 100):");
            for (int i = 0; i < 5; i++) {
                marks[i] = scanner.nextDouble();
            }

            // Creating Student object
            Student student = new Student(name, rollNo, marks);
```

```
            // Calculating average and grade
            student.calculateAverage();
            student.calculateGrade();

            // Displaying student information
            student.displayStudentInfo();

        } catch (InputMismatchException e) {
            System.out.println("Error: Please enter valid numeric values for roll number and marks.");
        } catch (IllegalArgumentException e) {
            System.out.println("Error: " + e.getMessage());
        } finally {
        }
    }
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 5>javac StudentDemo.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignment 5>java StudentDemo
Enter Student Name: sonu
Enter Roll Number: 081
Enter marks for 5 subjects (out of 100):
80
75
91
49
60

Student Details:
Name: sonu
Roll Number: 81
Marks: 80.0 75.0 91.0 49.0 60.0
Average Marks: 71.0
Grade: C
```