

Que 1. 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.Scanner;

import java.util.InputMismatchException;

class ElectricityBill{

    private String Name;

    private double unitsConsumed;


    public ElectricityBill(String Name,double unitsConsumed){

        this.Name= Name;

        this.unitsConsumed = unitsConsumed;

    }

    public double calculateBillAmount(){

        double Amount = 0;

        if(unitsConsumed <= 100){

            Amount = 5* unitsConsumed;

            return Amount;

        }

        if(unitsConsumed > 100 && unitsConsumed <= 300){

            Amount = 7 * unitsConsumed;

            return Amount;

        }

        if(unitsConsumed > 300){

            Amount = 10 * unitsConsumed;

            return Amount;

        }

        return Amount;

    }

}

```

```

class ElectricityBillCalculationEx{

```

```

public static void main(String[] args){

    Scanner sc = new Scanner(System.in);

    System.out.println("Enter name of customer= ");

    String Name = sc.next();

    double unitsConsumed = 0;

    try{

        System.out.println("Enter units consumed= ");

        unitsConsumed = sc.nextDouble();

        if(unitsConsumed < 0){

            throw new IllegalArgumentException("Please enter positive value");

        }

    }catch(InputMismatchException e){

        System.out.println("Please enter numeric data");

    }catch(IllegalArgumentException e){

        System.out.println("Please enter positive data");

    }

    ElectricityBill e = new ElectricityBill(Name, unitsConsumed);

    System.out.println("Bill Amount of = " +Name + " is "+e.calculateBillAmount());

}

}

```

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: $80 \leq \text{average} < 90$
 - C: $70 \leq \text{average} < 80$
 - D: $60 \leq \text{average} < 70$
 - 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.Scanner;

import java.util.InputMismatchException;

class Student{

    String name;

    int rollNo;

    double marks[] = new double[5];

    double average;

    char grade;

    public Student(String name,int rollNo, double marks[]){

        this.name=name;

        this.rollNo =rollNo;

        for(int i = 0;i<marks.length;i++){

            if(marks[i] < 0 || marks[i] > 100){

                throw new IllegalArgumentException("please enter valid marks");

            }

            this.marks[i] = marks [i];

        }

    }

    public void calculateAverage(){

        double total =0;

        for(int i =0;i < marks.length; i++){

            total += marks[i];

        }

        this.average =total/marks.length;

    }

}
```

```
public void calculateGrade(){  
    if(average >= 90){  
        grade='A';  
    }  
    if(average >= 80){  
        grade='B';  
    }  
    if(average >= 70){  
        grade = 'C';  
    }  
    if(average >= 60){  
        grade ='D';  
    }  
    if(average < 60){  
        grade = 'F';  
    }  
}
```

```
public void displayStudentInfo(){  
    System.out.println("Student Name= " +name);  
  
    System.out.println("Student roll number= " +rollNo);  
  
    System.out.println("marks = ");  
    for(int i =0;i<marks.length;i++){  
        System.out.print(" "marks[i]);  
    }  
    System.out.println("average = "+average);  
    System.out.println("grade = "+grade);  
}
```

```
}
```

```

class StudentMGCalculation{

    public static void main(String[] args){

        try{

            Scanner sc = new Scanner(System.in);

            System.out.println("Enter Student name= ");

            String name = sc.next();

            System.out.println("Enter Student roll number= ");

            int rollNo = sc.nextInt();

            System.out.println("Enter Marks of Student= ");


            double marks[] = new double[5];

            for(int i = 0; i < marks.length;i++){

                marks[i] = sc.nextDouble();


                if(marks[i] < 0 || marks[i] > 100){

                    throw new IllegalArgumentException("please enter valid marks");

                }

            }

            Student s = new Student(name,rollNo,marks);

            s.calculateAverage();

            s.calculateGrade();

            s.displayStudentInfo();

        }catch(InputMismatchException e){

            System.out.println("Invalid input! Please enter numeric values for roll number and marks.");

        }catch(IllegalArgumentException e){

            System.out.println("invalid marks ");

        }

    }

}

```