Name- Aarya Sanjay Dange

Assignment 05 (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:

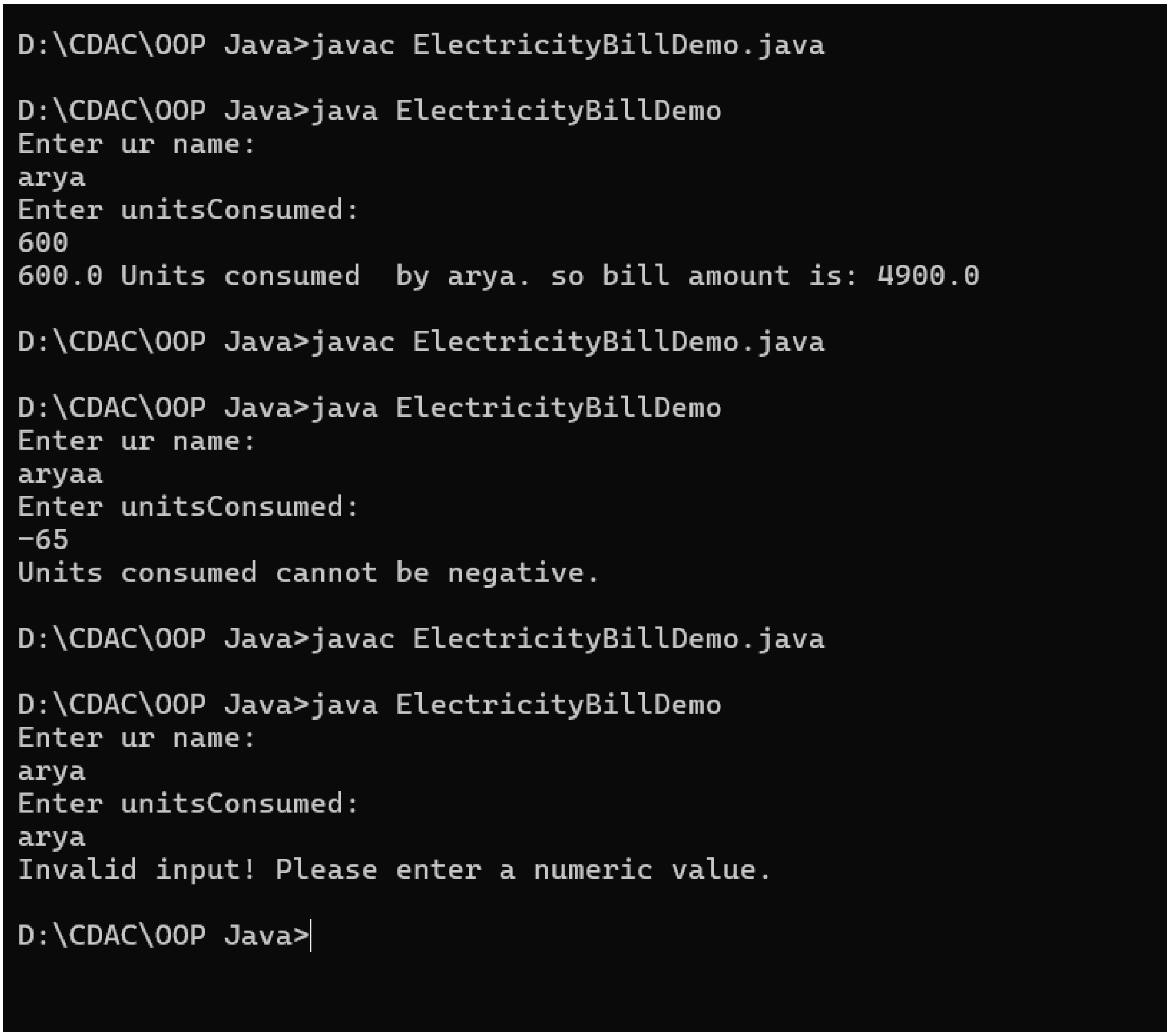
○ First 100 units: Rs. 5 per unit
○ Next 200 units (101–300): Rs. 7 per unit
○ Above 300 units: Rs. 10 per unit

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.

Ans:

```
import java.util.*;
class ElectricityBill{
	String customerName;
	double unitsConsumed;
	double billAmount;

	public ElectricityBill(String customerName,double unitsConsumed){
        if(unitsConsumed<0){
            throw new IllegalArgumentException("Units
consumed cannot be negative.");
        }
            this.customerName=customerName;
            this.unitsConsumed=unitsConsumed;


    }
    double calculateBillAmount(){
        if(unitsConsumed<=100){
            return (5*unitsConsumed) ;
        }
        else if(unitsConsumed>100 &&
unitsConsumed<=300){
            return ((100*5)+((unitsConsumed-100)*7)) ;
        }
        else{
            return
((100*5)+(200*7)+(unitsConsumed-300)*10) ;
        }
    }
}
class ElectricityBillDemo{
    public static void main(String[] args){
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter ur name: ");
    String customerName=sc.nextLine();
    while(true){
    try{
        System.out.println("Enter unitsConsumed: ");
        double unitsConsumed=sc.nextDouble();

        ElectricityBill b1=new
ElectricityBill(customerName,unitsConsumed);
        System.out.println(b1.unitsConsumed+" Units
consumed "+" by "+b1.customerName+". so bill amount is:
"+b1.calculateBillAmount());

    }catch(InputMismatchException e){
        System.out.println("Invalid input! Please enter a
numeric value.");
    }catch(IllegalArgumentException e){
        System.out.println("Units consumed cannot be
negative.");
    }
    break;
    }
    }
}
```



```
D:\CDAC\OOP Java>javac ElectricityBillDemo.java

D:\CDAC\OOP Java>java ElectricityBillDemo
Enter ur name:
arya
Enter unitsConsumed:
600
600.0 Units consumed  by arya. so bill amount is: 4900.0

D:\CDAC\OOP Java>javac ElectricityBillDemo.java

D:\CDAC\OOP Java>java ElectricityBillDemo
Enter ur name:
aryaa
Enter unitsConsumed:
-65
Units consumed cannot be negative.

D:\CDAC\OOP Java>javac ElectricityBillDemo.java

D:\CDAC\OOP Java>java ElectricityBillDemo
Enter ur name:
arya
Enter unitsConsumed:
arya
Invalid input! Please enter a numeric value.

D:\CDAC\OOP Java>
```

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

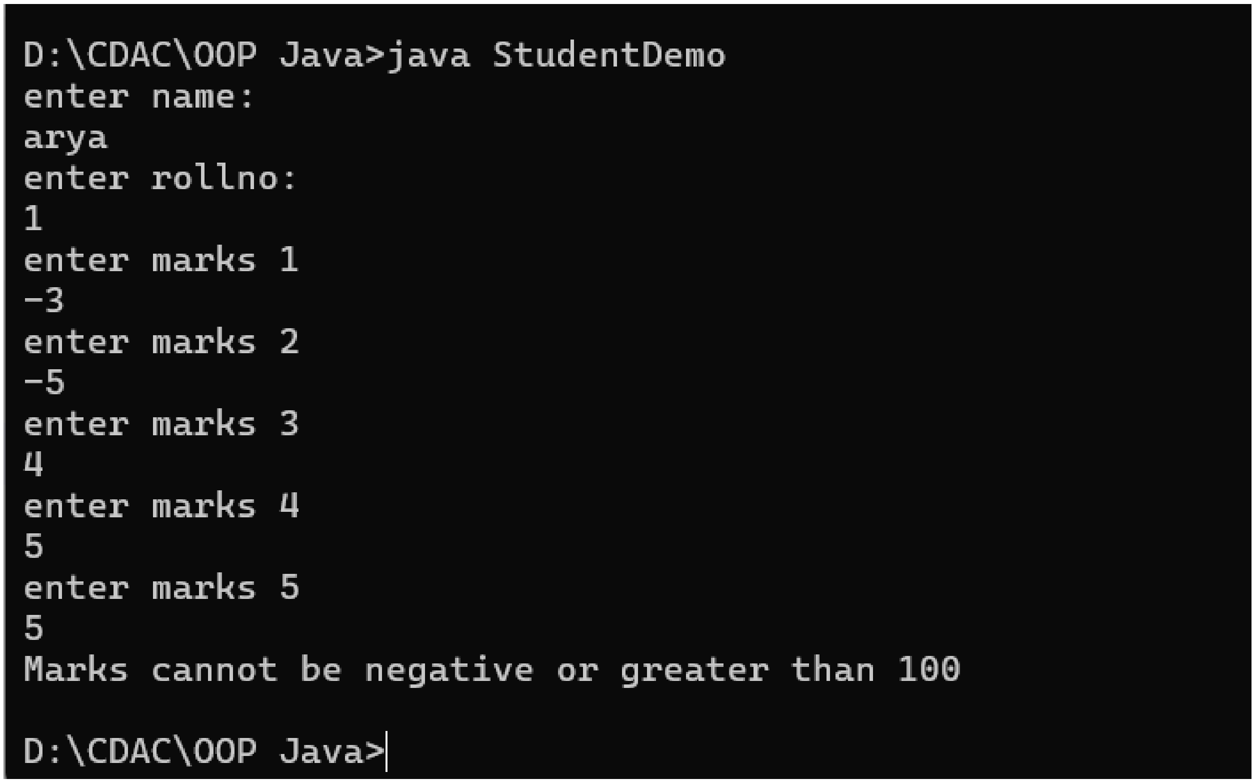
In the main() method:

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.

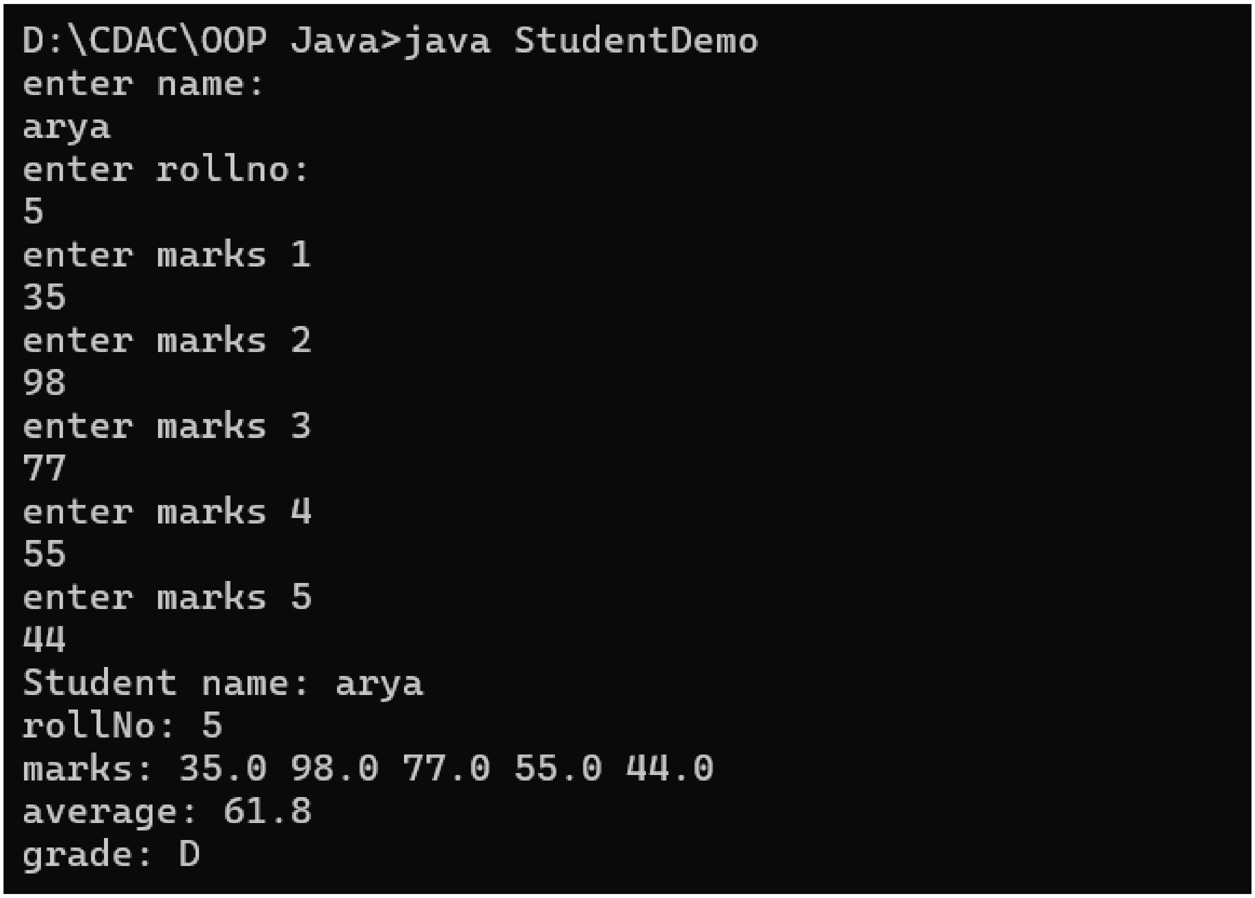
Ans:

```
import java.util.*;
class Student{
	String name;
	int rollNo;
	double marks[]=new double[5];
	double avg;
	char grade;


	public Student(String name,int rollNo,double[] marks){
		this.name=name;
		this.rollNo=rollNo;
		for(double mark:marks){
			if(mark<0 || mark>100){
```

```
                throw new
IllegalArgumentException("Marks cannot be negative or
greater than 100");
            }
        }
        this.marks=marks;
    }
    double calculateAverage(){
        double sum=0;
        for(double mark:marks){
            sum=sum+mark;
        }
        avg=sum/marks.length;
        return avg;
    }
    String mk(){
        StringBuilder sb = new StringBuilder();
  for (double mark : marks) {
    sb.append(mark).append(" ");
  }
  return sb.toString().trim();
    }
    char calculateGrade(){
        if(avg >=90){
            return 'A';
        }
        else if(avg >=80){
            return 'B';
        }
        else if(avg >=70){
            return 'C';
        }
        else if(avg >=60){
            return 'D';
        }
        else{
            return 'F';
        }
    }
    String displayStudentInfo(){
        return "Student name: "+name+"\nrollNo:
"+rollNo+"\nmarks: "+mk()+"\naverage:
"+calculateAverage()+"\ngrade: "+calculateGrade();
    }
}
class StudentDemo{
        public static void main(String args[]){

        Scanner sc=new Scanner(System.in);
        System.out.println("enter name:");
        String name=sc.nextLine();
        System.out.println("enter rollno:");
        int rollNo=sc.nextInt();

        try{
            double marks[]=new double[5];
            for(int i = 0; i < marks.length; i++) {
                  System.out.println("enter marks "+(i+1));
                  marks[i] = sc.nextDouble();
            }
            Student s1=new Student(name,rollNo,marks);
            System.out.println(s1.displayStudentInfo());
            }catch(IllegalArgumentException e){
                  System.out.println("Marks cannot be negative
or greater than 100");
            }


            }
}
```



```
D:\CDAC\OOP Java>java StudentDemo
enter name:
arya
enter rollno:
1
enter marks 1
-3
enter marks 2
-5
enter marks 3
4
enter marks 4
5
enter marks 5
5
Marks cannot be negative or greater than 100

D:\CDAC\OOP Java>
```



```
D:\CDAC\OOP Java>java StudentDemo
enter name:
arya
enter rollno:
5
enter marks 1
35
enter marks 2
98
enter marks 3
77
enter marks 4
55
enter marks 5
44
Student name: arya
rollNo: 5
marks: 35.0 98.0 77.0 55.0 44.0
average: 61.8
grade: D
```