

AI Assisted Coding

Assignment – 3.3

Name : Ch.Harish Rao

Ht.no : 2303A51661

Batch : 23

Prompt – 1 : Use an AI tool to generate a Java program that reads previous units, current units, and customer type, validates input, calculates units consumed, and displays the result using simple conditional logic without functions.

Code :

```
import java.util.Scanner;

public class main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter previous units: ");

        int previousUnits = scanner.nextInt();

        // Input current units

        System.out.print("Enter current units: ");

        int currentUnits = scanner.nextInt();

        System.out.print("Enter customer type (R for Residential, C for Commercial, I for Industrial): ");

        char customerType = scanner.next().toUpperCase().charAt(0);

        if (previousUnits < 0 || currentUnits < 0 || currentUnits < previousUnits ||

            (customerType != 'R' && customerType != 'C' && customerType != 'I')) {

            System.out.println("Invalid input. Please check the values entered.");

            return;

        }

        int unitsConsumed = currentUnits - previousUnits;

        System.out.println("Customer Type: " +
```

```

        (customerType == 'R' ? "Residential" :
        customerType == 'C' ? "Commercial" : "Industrial"));

    System.out.println("Units Consumed: " + unitsConsumed);
}
}

```

Output :

Enter previous units: 2200

Enter current units: 2500

Enter customer type (R for Residential, C for Commercial, I for Industrial): C

Customer Type: Commercial

Units Consumed: 300

Task – 2 Prompt : Extend the java program from Task 1 to calculate energy charges using conditional logic based on domestic, commercial, and industrial consumers, and optimize the code using AI-assisted prompts.

Code :

```

import java.util.Scanner;

public class lab03 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Input previous units

        System.out.print("Enter previous units: ");

        int previousUnits = scanner.nextInt();

        // Input current units

        System.out.print("Enter current units: ");

        int currentUnits = scanner.nextInt();

        // Input customer type

        System.out.print("Enter customer type (R for Residential, C for Commercial, I for Industrial): ");
    }
}

```

```

char customerType = scanner.next().toUpperCase().charAt(0);

// Validate input
if (previousUnits < 0 || currentUnits < 0 || currentUnits < previousUnits ||
    (customerType != 'R' && customerType != 'C' && customerType != 'I')) {
    System.out.println("Invalid input. Please check the values entered.");
    return;
}

// Calculate units consumed
int unitsConsumed = currentUnits - previousUnits;

// Calculate energy charges
double energyCharge = 0.0;
if (customerType == 'R') { // Residential
    if (unitsConsumed <= 100) {
        energyCharge = unitsConsumed * 0.5;
    } else if (unitsConsumed <= 300) {
        energyCharge = 100 * 0.5 + (unitsConsumed - 100) * 0.75;
    } else {
        energyCharge = 100 * 0.5 + 200 * 0.75 + (unitsConsumed - 300) * 1.0;
    }
} else if (customerType == 'C') { // Commercial
    if (unitsConsumed <= 200) {
        energyCharge = unitsConsumed * 0.75;
    } else {
        energyCharge = 200 * 0.75 + (unitsConsumed - 200) * 1.25;
    }
} else if (customerType == 'I') { // Industrial
    if (unitsConsumed <= 500) {
        energyCharge = unitsConsumed * 1.0;
    }
}

```

```

        } else {
            energyCharge = 500 * 1.0 + (unitsConsumed - 500) * 1.5;
        }
    }

    // Display result
    System.out.println("Customer Type: " +
        (customerType == 'R' ? "Residential" :
        customerType == 'C' ? "Commercial" : "Industrial"));
    System.out.println("Units Consumed: " + unitsConsumed);
    System.out.println("Energy Charge: $" + energyCharge);
}
}

```

Output :

Enter previous units: 2000

Enter current units: 2500

Enter customer type (R for Residential, C for Commercial, I for Industrial): R

Customer Type: Residential

Units Consumed: 500

Energy Charge: \$400.0

Task -3 Prompt :

Use AI tools to design a modular Java electricity billing program that calculates energy charges and fixed charges using user-defined functions and returns computed values to the main program.

Code :

```

import java.util.Scanner;

public class lab03 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Input previous units
    }
}

```

```

System.out.print("Enter previous units: ");

int previousUnits = scanner.nextInt();


// Input current units

System.out.print("Enter current units: ");

int currentUnits = scanner.nextInt();


// Input customer type

System.out.print("Enter customer type (R for Residential, C for Commercial, I for Industrial): ");

char customerType = scanner.next().toUpperCase().charAt(0);


// Validate input

if (previousUnits < 0 || currentUnits < 0 || currentUnits < previousUnits ||
    (customerType != 'R' && customerType != 'C' && customerType != 'I')) {
    System.out.println("Invalid input. Please check the values entered.");
    return;
}


// Calculate units consumed

int unitsConsumed = calculateUnitsConsumed(previousUnits, currentUnits);


// Calculate energy charges and fixed charges

double energyCharge = calculateEnergyCharge(unitsConsumed, customerType);

double fixedCharge = calculateFixedCharge(customerType);


// Display result

System.out.println("Customer Type: " +
    (customerType == 'R' ? "Residential" :
    customerType == 'C' ? "Commercial" : "Industrial"));

System.out.println("Units Consumed: " + unitsConsumed);

System.out.printf("Energy Charge: $%.2f%n", energyCharge);

```

```

        System.out.printf("Fixed Charge: $%.2f%n", fixedCharge);
        System.out.printf("Total Charge: $%.2f%n", (energyCharge + fixedCharge));
    }

    public static int calculateUnitsConsumed(int previousUnits, int currentUnits) {
        return currentUnits - previousUnits;
    }

    public static double calculateEnergyCharge(int unitsConsumed, char customerType) {
        double charge = 0.0;

        if (customerType == 'R') { // Residential
            if (unitsConsumed <= 100) {
                charge = unitsConsumed * 0.5;
            } else if (unitsConsumed <= 300) {
                charge = 100 * 0.5 + (unitsConsumed - 100) * 0.75;
            } else {
                charge = 100 * 0.5 + 200 * 0.75 + (unitsConsumed - 300) * 1.0;
            }
        } else if (customerType == 'C') { // Commercial
            if (unitsConsumed <= 200) {
                charge = unitsConsumed * 0.75;
            } else {
                charge = 200 * 0.75 + (unitsConsumed - 200) * 1.25;
            }
        } else if (customerType == 'I') { // Industrial
            if (unitsConsumed <= 500) {
                charge = unitsConsumed * 1.0;
            } else {
                charge = 500 * 1.0 + (unitsConsumed - 500) * 1.5;
            }
        }

        return charge;
    }
}

```

```

public static double calculateFixedCharge(char customerType) {
    switch (customerType) {
        case 'R':
            return 5.0; // Fixed charge for Residential
        case 'C':
            return 10.0; // Fixed charge for Commercial
        case 'I':
            return 20.0; // Fixed charge for Industrial
        default:
            return 0.0;
    }
}

```

Output :

Enter previous units: 2000

Enter current units: 2500

Enter customer type (R for Residential, C for Commercial, I for Industrial): C

Customer Type: Commercial

Units Consumed: 500

Energy Charge: \$525.00

Fixed Charge: \$10.00

Total Charge: \$535.00

Task - 4 Prompt : Extend the Java electricity billing program to calculate fixed charges, customer charges, and electricity duty using AI-assisted modular methods and display each component clearly.

Code :

```

import java.util.Scanner;

public class lab03 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
    }
}

```

```

// Input previous units
System.out.print("Enter previous units: ");
int previousUnits = scanner.nextInt();

// Input current units
System.out.print("Enter current units: ");
int currentUnits = scanner.nextInt();

// Input customer type
System.out.print("Enter customer type (R for Residential, C for Commercial, I for Industrial): ");
char customerType = scanner.next().toUpperCase().charAt(0);

// Validate input
if (previousUnits < 0 || currentUnits < 0 || currentUnits < previousUnits ||
    (customerType != 'R' && customerType != 'C' && customerType != 'I')) {
    System.out.println("Invalid input. Please check the values entered.");
    return;
}

// Calculate units consumed
int unitsConsumed = calculateUnitsConsumed(previousUnits, currentUnits);

// Calculate energy charges, fixed charges, customer charges, and electricity duty
double energyCharge = calculateEnergyCharge(unitsConsumed, customerType);
double fixedCharge = calculateFixedCharge(customerType);
double customerCharge = calculateCustomerCharge(customerType);
double electricityDuty = calculateElectricityDuty(energyCharge);

// Display result
System.out.println("Customer Type: " +
    (customerType == 'R' ? "Residential" :

```



```

        customerType == 'C' ? "Commercial" : "Industrial"));

System.out.println("Units Consumed: " + unitsConsumed);

System.out.printf("Energy Charge: $%.2f%n", energyCharge);

System.out.printf("Fixed Charge: $%.2f%n", fixedCharge);

System.out.printf("Customer Charge: $%.2f%n", customerCharge);

System.out.printf("Electricity Duty: $%.2f%n", electricityDuty);

System.out.printf("Total Charge: $%.2f%n", (energyCharge + fixedCharge + customerCharge +
electricityDuty));

}

```

```

public static int calculateUnitsConsumed(int previousUnits, int currentUnits) {

    return currentUnits - previousUnits;

}

```

```

public static double calculateEnergyCharge(int unitsConsumed, char customerType) {

    double charge = 0.0;

    if (customerType == 'R') { // Residential

        if (unitsConsumed <= 100) {

            charge = unitsConsumed * 0.5;

        } else if (unitsConsumed <= 300) {

            charge = 100 * 0.5 + (unitsConsumed - 100) * 0.75;

        } else {

            charge = 100 * 0.5 + 200 * 0.75 + (unitsConsumed - 300) * 1.0;

        }

    } else if (customerType == 'C') { // Commercial

        if (unitsConsumed <= 200) {

            charge = unitsConsumed * 0.75;

        } else {

            charge = 200 * 0.75 + (unitsConsumed - 200) * 1.25;

        }

    } else if (customerType == 'I') { // Industrial

```

```

    if (unitsConsumed <= 500) {
        charge = unitsConsumed * 1.0;
    } else {
        charge = 500 * 1.0 + (unitsConsumed - 500) * 1.5;
    }
}
return charge;
}

```

```

public static double calculateFixedCharge(char customerType) {
    switch (customerType) {
        case 'R':
            return 5.0; // Fixed charge for Residential
        case 'C':
            return 10.0; // Fixed charge for Commercial
        case 'I':
            return 20.0; // Fixed charge for Industrial
        default:
            return 0.0;
    }
}

```

```

public static double calculateCustomerCharge(char customerType) {
    switch (customerType) {
        case 'R':
            return 1.5; // Customer charge for Residential
        case 'C':
            return 2.5; // Customer charge for Commercial
        case 'I':
            return 3.5; // Customer charge for Industrial
        default:

```

```

        return 0.0;
    }
}

public static double calculateElectricityDuty(double energyCharge) {
    return energyCharge * 0.15; // Electricity duty is 15% of energy charge
}
}

```

Output :

Enter previous units: 2000

Enter current units: 2500

Enter customer type (R for Residential, C for Commercial, I for Industrial): R

Customer Type: Residential

Units Consumed: 500

Energy Charge: \$400.00

Fixed Charge: \$5.00

Customer Charge: \$1.50

Electricity Duty: \$60.00

Total Charge: \$466.50

Task -5 Prompt : Develop the final Java electricity billing program to calculate and display all bill components and the total bill amount, and analyze it for accuracy, readability, and real-world applicability.

Code :

```

import java.util.Scanner;

public class lab03 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Input previous units

        System.out.print("Enter previous units: ");

        int previousUnits = scanner.nextInt();
    }
}

```

```

// Input current units

System.out.print("Enter current units: ");

int currentUnits = scanner.nextInt();


// Input customer type

System.out.print("Enter customer type (R for Residential, C for Commercial, I for Industrial): ");

char customerType = scanner.next().toUpperCase().charAt(0);


// Validate input

if (previousUnits < 0 || currentUnits < 0 || currentUnits < previousUnits ||
    (customerType != 'R' && customerType != 'C' && customerType != 'I')) {
    System.out.println("Invalid input. Please check the values entered.");
    return;
}


// Calculate units consumed

int unitsConsumed = calculateUnitsConsumed(previousUnits, currentUnits);


// Calculate energy charges, fixed charges, customer charges, and electricity duty

double energyCharge = calculateEnergyCharge(unitsConsumed, customerType);
double fixedCharge = calculateFixedCharge(customerType);
double customerCharge = calculateCustomerCharge(customerType);
double electricityDuty = calculateElectricityDuty(energyCharge);


// Display result

System.out.println("Customer Type: " +
    (customerType == 'R' ? "Residential" :
    customerType == 'C' ? "Commercial" : "Industrial"));

System.out.println("Units Consumed: " + unitsConsumed);

System.out.printf("Energy Charge: $%.2f%n", energyCharge);

```

```
System.out.printf("Fixed Charge: $%.2f%n", fixedCharge);  
System.out.printf("Customer Charge: $%.2f%n", customerCharge);  
System.out.printf("Electricity Duty: $%.2f%n", electricityDuty);  
System.out.printf("Total Charge: $%.2f%n", (energyCharge + fixedCharge + customerCharge +  
electricityDuty));  
}
```

```
public static int calculateUnitsConsumed(int previousUnits, int currentUnits) {  
    return currentUnits - previousUnits;  
}
```

```
public static double calculateEnergyCharge(int unitsConsumed, char customerType) {  
    double charge = 0.0;  
    if (customerType == 'R') { // Residential  
        if (unitsConsumed <= 100) {  
            charge = unitsConsumed * 0.5;  
        } else if (unitsConsumed <= 300) {  
            charge = 100 * 0.5 + (unitsConsumed - 100) * 0.75;  
        } else {  
            charge = 100 * 0.5 + 200 * 0.75 + (unitsConsumed - 300) * 1.0;  
        }  
    } else if (customerType == 'C') { // Commercial  
        if (unitsConsumed <= 200) {  
            charge = unitsConsumed * 0.75;  
        } else {  
            charge = 200 * 0.75 + (unitsConsumed - 200) * 1.25;  
        }  
    } else if (customerType == 'I') { // Industrial  
        if (unitsConsumed <= 500) {  
            charge = unitsConsumed * 1.0;  
        } else {  

```

```

        charge = 500 * 1.0 + (unitsConsumed - 500) * 1.5;
    }
}
return charge;
}

public static double calculateFixedCharge(char customerType) {
    switch (customerType) {
        case 'R':
            return 5.0; // Fixed charge for Residential
        case 'C':
            return 10.0; // Fixed charge for Commercial
        case 'I':
            return 20.0; // Fixed charge for Industrial
        default:
            return 0.0;
    }
}

public static double calculateCustomerCharge(char customerType) {
    switch (customerType) {
        case 'R':
            return 1.5; // Customer charge for Residential
        case 'C':
            return 2.5; // Customer charge for Commercial
        case 'I':
            return 3.5; // Customer charge for Industrial
        default:
            return 0.0;
    }
}

public static double calculateElectricityDuty(double energyCharge) {
    return energyCharge * 0.15; // Electricity duty is 15% of energy charge
}

```

}

}

Output :

Enter previous units: 3000

Enter current units: 3500

Enter customer type (R for Residential, C for Commercial, I for Industrial): I

Customer Type: Industrial

Units Consumed: 500

Energy Charge: \$500.00

Fixed Charge: \$20.00

Customer Charge: \$3.50

Electricity Duty: \$75.00

Total Charge: \$598.50