Assignment Number: 3.3

NAME: N. NandaMukesh

H.T.NO: 2303A51410

BATCH: 24

Task 1: AI-Generated Logic for Reading Consumer Details

Scenario

An electricity billing system must collect accurate consumer data.

Task Description

Use an AI tool (GitHub Copilot / Gemini) to generate a Python program that:

• Reads:

o Previous Units (PU)

o Current Units (CU)

o Type of Customer

• Calculates units consumed

• Implements logic directly in the main program (no functions)

```
53  # application for TGNPDCL-electricity bill generation take 3 user inputs,
54  pervious_inputs = float(input("Enter Previous Month's Meter Reading: "))
55  current_inputs = float(input("Enter Current Month's Meter Reading: "))
56  units_consumed = current_inputs - pervious_inputs
57  customer_type = input("Enter Customer Type (Domestic/Commercial/Industrial
58  if current_inputs < pervious_inputs:
59      print("Error: Current reading cannot be less than previous reading.")
60      exit()
61  units_consumed = current_inputs - pervious_inputs
62  print("Units Consumed:", units_consumed)
63
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter Previous Month's Meter Reading: 20
Enter Current Month's Meter Reading: 35
Enter Previous Month's Meter Reading: 20
Enter Current Month's Meter Reading: 35
Enter Current Month's Meter Reading: 35
Enter Customer Type (Domestic/Commercial/Industrial): commercial
Enter Customer Type (Domestic/Commercial/Industrial): commercial
Units Consumed: 15.0
Units Consumed: 15.0
```

Expected Output

• Correct input reading

• Units consumed calculation

• Screenshot showing AI-generated code

• Sample input and output

Task 2: Energy Charges Calculation Based on Units Consumed

Scenario

Energy charges depend on the number of units consumed and customer type.

Task Description

Review the AI-generated code from Task 1 and extend it to:

• Calculate Energy Charges (EC)

• Use conditional statements based on:

o Domestic

o Commercial

o Industrial consumers

• Improve readability using AI prompts such as:

o “Simplify energy charge calculation logic”

o “Optimize conditional statements”

Expected Output

• Correct EC calculation

• Clear conditional logic

```python
#write a function for Energy charges calculation based on customer type, and optimal code
def calculate_energy_charges(units_consumed, customer_type):
    if customer_type.lower() == "domestic":
        if units_consumed <= 100:
            charges = units_consumed * 1.5
        elif units_consumed <= 300:
            charges = (100 * 1.5) + ((units_consumed - 100) * 2.5)
        else:
            charges = (100 * 1.5) + (200 * 2.5) + ((units_consumed - 300) * 4.0)
    elif customer_type.lower() == "commercial":
        if units_consumed <= 100:
            charges = units_consumed * 2.0
        elif units_consumed <= 300:
            charges = (100 * 2.0) + ((units_consumed - 100) * 3.5)
        else:
            charges = (100 * 2.0) + (200 * 3.5) + ((units_consumed - 300) * 5.0)
    elif customer_type.lower() == "industrial":
        if units_consumed <= 100:
            charges = units_consumed * 3.0
        elif units_consumed <= 300:
            charges = (100 * 3.0) + ((units_consumed - 100) * 4.5)
        else:
            charges = (100 * 3.0) + (200 * 4.5) + ((units_consumed - 300) * 6.0)
    else:
        raise ValueError("Invalid customer type")
    return charges
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter Previous Month's Meter Reading: 34.6
Enter Current Month's Meter Reading: 45
Enter Customer Type (Domestic/Commercial/Industrial): commercial
Enter Current Month's Meter Reading: 45
Enter Current Month's Meter Reading: 45
Enter Customer Type (Domestic/Commercial/Industrial): commercial
Units Consumed: 10.399999999999999
Energy Charges for commercial customer: 20.799999999999997
```

• Original and improved versions (optional)

• Sample execution results

Task 3: Modular Design Using AI Assistance (Using Functions)

Scenario

Billing logic must be reusable for multiple consumers.

Task Description

Use AI assistance to generate a Python program that:

• Uses user-defined functions to:

o Calculate Energy Charges

o Calculate Fixed Charges

• Returns calculated values

• Includes meaningful comments

Expected Output

• Function-based Python program

• Correct EC and FC values

• Screenshots of AI-assisted function generation

• Test cases with output

```python
#write a well-commented python optimal code to calculate fixed charges based on consumed units and customer typ
def calculate_fixed_charges(units_consumed, customer_type):
    # Define fixed charges based on customer type
    fixed_charges = 0
    if customer_type.lower() == "domestic":
        if units_consumed <= 100:
            fixed_charges = 50
        elif units_consumed <= 300:
            fixed_charges = 100
        else:
            fixed_charges = 150
    elif customer_type.lower() == "commercial":
        if units_consumed <= 100:
            fixed_charges = 100
        elif units_consumed <= 300:
            fixed_charges = 200
        else:
            fixed_charges = 300
    elif customer_type.lower() == "industrial":
        if units_consumed <= 100:
            fixed_charges = 150
        elif units_consumed <= 300:
            fixed_charges = 300
        else:
            fixed_charges = 450
    return fixed_charges
fixed_charges = calculate_fixed_charges(units_consumed, customer_type)
print("Fixed Charges for", customer_type, "customer:", fixed_charges)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Enter Previous Month's Meter Reading: 45
Enter Current Month's Meter Reading: 356
Enter Customer Type (Domestic/Commercial/Industrial): domestic
Units Consumed: 311.0
Energy Charges for domestic customer: 694.0
Fixed Charges for domestic customer: 150
```

Task 4: Calculation of Additional Charges

Scenario

Electricity bills include multiple additional charges.

Task Description

Extend the program to calculate:

• FC – Fixed Charges

• CC – Customer Charges

• ED – Electricity Duty (percentage of EC)

Use AI prompts like:

• “Add electricity duty calculation”

• “Improve billing accuracy”

Expected Output

• Individual charge values printed

• Correct duty calculation

• Well-structured output

• Verified intermediate results

```
123
124   #extend the above code to calculate customer charges and add electricity duty calculation to improve billing accuracy withoutcreating functions
125   customer_charges = 20
126   electricity_duty = 0.05
127   total_charges = charges + fixed_charges + customer_charges
128   electricity_duty_amount = total_charges * electricity_duty
129   print("Total Charges:", total_charges)
130   print("Electricity Duty Amount:", electricity_duty_amount)
131   print("Final Bill Amount:", total_charges + electricity_duty_amount)
132
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\yadav\OneDrive\Desktop\AI-LAB> & C:/Users/yadav/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/yadav/OneDrive/Desktop/AI-LAB/assign.py
Enter Previous Month's Meter Reading: 20
Enter Current Month's Meter Reading: 30
Enter Customer Type (Domestic/Commercial/Industrial): commercial
Units Consumed: 10.0
Energy Charges for commercial customer: 20.0
Fixed Charges for commercial customer: 100
Total Charges: 140.0
Electricity Duty Amount: 7.0
Final Bill Amount: 147.0
PS C:\Users\yadav\OneDrive\Desktop\AI-LAB>
```

Task 5: Final Bill Generation and Output Analysis

Scenario

The final electricity bill must present all values clearly.

Task Description

Develop the final Python application to:

• Calculate total bill:

• Total Bill = EC + FC + CC + ED

• Display:

o Energy Charges (EC)

o Fixed Charges (FC)

o Customer Charges (CC)

o Electricity Duty (ED)

o Total Bill Amount

• Analyze the program based on:

o Accuracy

o Readability

o Real-world applicability

Expected Output

• Complete electricity bill output

• Neatly formatted display

• Sample input/output

• Short analysis paragraph