ASSIGNMENT-3.3
VELPULA VIGNESH
23O3A51707
BATCH-24

Lab 3: Application for TGNPDCL – Electricity Bill Generation Using Python & AI Tools

Lab Objectives

- To design a real-world electricity billing application using Python
- To use AI-assisted coding tools for logic generation and optimization
- To understand conditional logic and arithmetic operations
- To generate structured billing output similar to utility bills

Lab Outcomes (LOs)

After completing this lab, students will be able to:

- Read and validate user input in Python
- Apply conditional logic for tariff-based billing
- Use AI tools to assist in program development
- Calculate and display electricity bill components
- Build a complete real-time application

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)

Expected Output

- Correct input reading
- Units consumed calculation
- Screenshot showing AI-generated code
- Sample input and output.

TASK-1:

```python
#Task-1
#An electricity billing system must collect accurate consumer data.
#The system should be able to calculate the total bill based on the units consumed and the rate per units like previous Units(pu),current units(cu)and type of customer(tc) as input.
#it should Calculates units consumed.
#Implements logic directly in the main program (no functions)
#Displays the total bill amount.
#Assume the rate per unit is as follows:
#Residential: $0.12 per unit
#Commercial: $0.15 per unit
# Get user input
pu = float(input("Enter previous units (pu): "))
cu = float(input("Enter current units (cu): "))
#use
tc = input("Enter type of customer (residential/commercial): ").strip().lower()
# Calculate units consumed
units_consumed = cu - pu
# Determine rate per unit based on customer type
if tc == "residential":
    rate_per_unit = 0.12
elif tc == "commercial":
    rate_per_unit = 0.15
else:
    print("Invalid customer type. Please enter 'residential' or 'commercial'.")
    exit()
# Calculate total bill amount
total_bill = units_consumed * rate_per_unit
# Display the total bill amount
print(f"Total bill amount for {units_consumed} units consumed is: ${total_bill:.2f}")
```

OUTPUT:

```
vighn@Vighnesh MINGW64 ~/OneDrive/Desktop/AI ASSISTING CODING (master)
$ python assignment_3.3.py
Enter previous units (pu): 29
Enter current units (cu): 2.444
Enter type of customer (residential/commercial): commercial
Total bill amount for -26.556 units consumed is: $-3.98
```

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
- Original and improved versions (optional)

• Sample execution results

INPUT:

```python
#Task-2
#display a optimal python program to Energy charges depend on the number of units consumed and customer type.
#Use conditional statements based on(Domestic,Commercial,Industrial consumers)
# Get user input
units = float(input("Enter the number of units consumed: "))
customer_type = input("Enter customer type (domestic/commercial/industrial): ").strip().lower()
# Determine rate per unit based on customer type
if customer_type == "domestic":
    if units <= 100:
        rate_per_unit = 0.10
    elif units <= 300:
        rate_per_unit = 0.15
    else:
        rate_per_unit = 0.20
elif customer_type == "commercial":
    if units <= 100:
        rate_per_unit = 0.15
    elif units <= 300:
        rate_per_unit = 0.20
    else:
        rate_per_unit = 0.25
elif customer_type == "industrial":
    if units <= 100:
        rate_per_unit = 0.20
    elif units <= 300:
        rate_per_unit = 0.25
    else:
        rate_per_unit = 0.30
else:
    print("Invalid customer type. Please enter 'domestic', 'commercial', or 'industrial'.")
    exit()
# Calculate total energy charges
total_charges = units * rate_per_unit
# Display the total energy charges
print(f"Total energy charges for {units} units consumed is: ${total_charges:.2f}")
```

OUTPUT:

```
Enter the number of units consumed: 234
Enter customer type (domestic/commercial/industrial): domestic
Total energy charges for 234.0 units consumed is: $35.10
Task 2 completed.
```

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 outputs

INPUT:

```python
#Task-3
#display a optimal python program to Billing logic must be reusable for multiple consumers by using functions.
#it should use the user-defined functions to calculate energy charges and calculate fixed charges.
#it should return calculated values.
#includes meaningful comments.
# Function to calculate energy charges based on units consumed and customer type
def calculate_energy_charges(units, customer_type):
    """Calculate energy charges based on units consumed and customer type."""
    if customer_type == "domestic":
        if units <= 100:
            rate_per_unit = 0.10
        elif units <= 300:
            rate_per_unit = 0.15
        else:
            rate_per_unit = 0.20
    elif customer_type == "commercial":
        if units <= 100:
            rate_per_unit = 0.15
        elif units <= 300:
            rate_per_unit = 0.20
        else:
            rate_per_unit = 0.25
    elif customer_type == "industrial":
        if units <= 100:
            rate_per_unit = 0.20
        elif units <= 300:
            rate_per_unit = 0.25
        else:
            rate_per_unit = 0.30
    else:
        raise ValueError("Invalid customer type. Please enter 'domestic', 'commercial', or 'industrial'.")

    return units * rate_per_unit
# Function to calculate fixed charges based on customer type
def calculate_fixed_charges(customer_type):
    """Calculate fixed charges based on customer type."""
    if customer_type == "domestic":
        return 50.0
    elif customer_type == "commercial":
        return 100.0
    elif customer_type == "industrial":
        return 150.0
    else:
        raise ValueError("Invalid customer type. Please enter 'domestic', 'commercial', or 'industrial'.")
```

```python
# Example usage
units_consumed = float(input("Enter the number of units consumed: "))
customer_type = input("Enter customer type (domestic/commercial/industrial): ").strip().lower()
try:
    energy_charges = calculate_energy_charges(units_consumed, customer_type)
    fixed_charges = calculate_fixed_charges(customer_type)
    total_bill = energy_charges + fixed_charges
    print(f"Total bill amount for {units_consumed} units consumed is: ${total_bill:.2f}")
except ValueError as e:
    print(e)
print("Task 3 completed.")
```

OUTPUT:

```
Enter the number of units consumed: 23
Enter customer type (domestic/commercial/industrial): domestic
Total bill amount for 23.0 units consumed is: $52.30
Task 3 completed.
```

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

INPUT:

```python
#Task-4
#display a python program to Electricity bills include multiple additional charges.
#Extend the program to calculate(FC – Fixed Charges,CC – Customer Charges,ED – Electricity Duty (percentage of EC).
#Get user input
#the output should be Individual charge values printed,Correct duty calculation,Well-structured output,Verified intermediate results
def calculate_total_bill(units, customer_type, ed_rate):
    """Calculate total electricity bill including additional charges."""
    # Calculate energy charges
    energy_charges = calculate_energy_charges(units, customer_type)
    # Calculate fixed charges
    fixed_charges = calculate_fixed_charges(customer_type)
    # Calculate electricity duty
    electricity_duty = (ed_rate / 100) * energy_charges
    # Calculate total bill
    total_bill = energy_charges + fixed_charges + electricity_duty
    return energy_charges, fixed_charges, electricity_duty, total_bill
# Example usage
units_consumed = float(input("Enter the number of units consumed: "))
customer_type = input("Enter customer type (domestic/commercial/industrial): ").strip().lower()
ed_rate = float(input("Enter Electricity Duty rate (as a percentage): "))
try:
    energy_charges, fixed_charges, electricity_duty, total_bill = calculate_total_bill(units_consumed, customer_type, ed_rate)
    print(f"Energy Charges: ${energy_charges:.2f}")
    print(f"Fixed Charges: ${fixed_charges:.2f}")
    print(f"Electricity Duty (@{ed_rate}%): ${electricity_duty:.2f}")
    print(f"Total Bill Amount: ${total_bill:.2f}")
except ValueError as e:
    print(e)
```

OUTPUT:

```
PS C:\Users\VIGNESH\OneDrive\Desktop\AI ASSESTENT CODING> & C:\Users\VIGNESH\AppData\Local\Microsoft\WindowsApps\python3.13.exe "c:/Users/VIGNESH/O
neDrive/Desktop/AI ASSESTENT CODING/Assingment-3.3.py"
Enter previous units: 10
Enter current units: 20
Enter customer type (Domestic/Commercial/Industrial): domestic
Units consumed: 10.0
Energy charges for domestic customer: 15.0
Fixed charges for domestic customer: 50
Total customer charges: 85.0
Electricity duty: 4.25
```

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

INPUT:

```python
#Task-5
#display a optimal python program to The final electricity bill must present all values clearly.
#develop the final Python application to Calculate total bill(Total Bill = EC + FC + CC + ED).
#It should display Energy Charges (EC), Fixed Charges (FC), Customer Charges (CC), Electricity Duty (ED), and Total Bill Amount.
#The output should be Complete electricity bill,Neatly formatted display with short Short analysis paragraph.
def display_electricity_bill(units, customer_type, ed_rate):
    """Display the complete electricity bill with all charges."""
    energy_charges, fixed_charges, electricity_duty, total_bill = calculate_total_bill(units, customer_type, ed_rate)

    print("\n--- Electricity Bill ---")
    print(f"Units Consumed: {units} units")
    print(f"Customer Type: {customer_type.capitalize()}")
    print(f"Energy Charges (EC): ${energy_charges:.2f}")
    print(f"Fixed Charges (FC): ${fixed_charges:.2f}")
    print(f"Electricity Duty (ED) @ {ed_rate}%: ${electricity_duty:.2f}")
    print(f"Total Bill Amount: ${total_bill:.2f}")
    print("------------------------")
    print("Analysis: The electricity bill is calculated based on the units consumed and the type of customer. "
          "The energy charges vary according to consumption slabs, while fixed charges are constant for each customer type. "
          "Electricity duty is applied as a percentage of the energy charges, contributing to the overall bill amount.")

print("Task 5 completed.")
```

OUTPUT:

```
----- Bill Details -----
Energy Charges: 15.0
Fixed Charges: 50
Customer Charges: 20
Electricity Duty: 4.25
Total Bill Amount: 154.25
```