

ASSIGNMENT 3.3

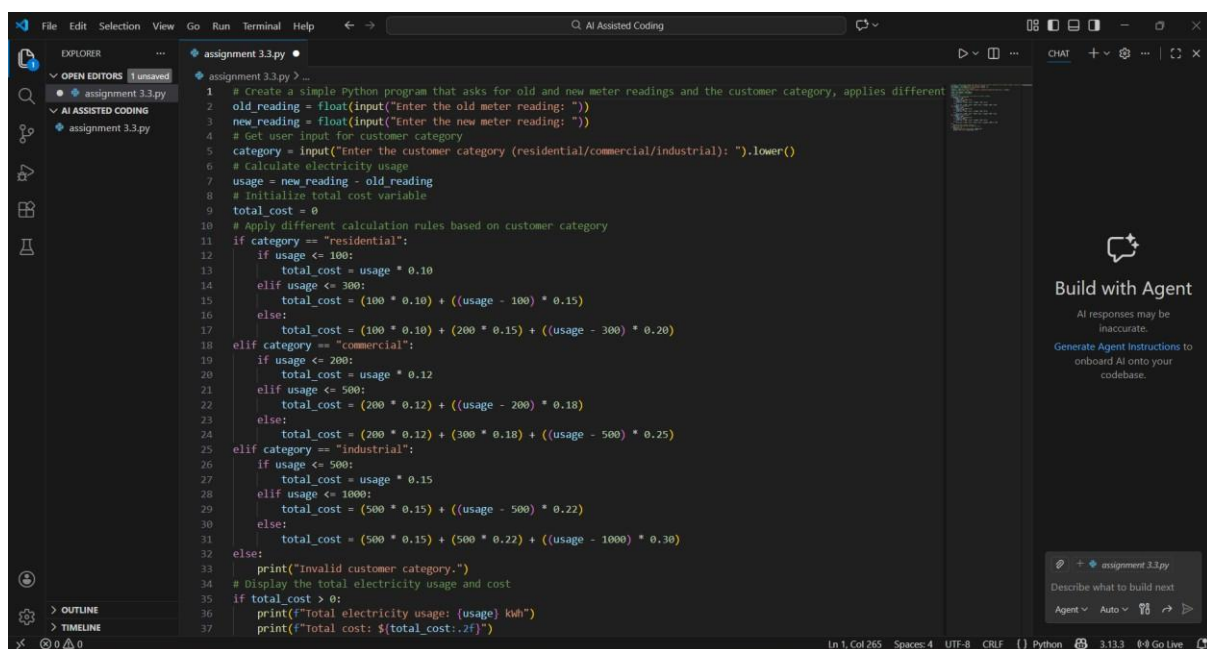
2303A51314

Batch – 05

Task – 01

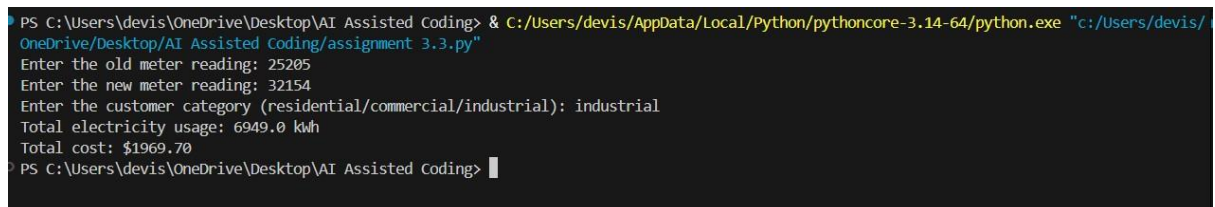
Prompt : Create a simple Python program that asks for old and new meter readings and the customer category, applies different calculation rules for each category, and displays the total electricity usage. Write all logic directly in the main code without using functions.

Code :



```
1 # Create a simple Python program that asks for old and new meter readings and the customer category, applies different
2 old_reading = float(input("Enter the old meter reading: "))
3 new_reading = float(input("Enter the new meter reading: "))
4 # Get user input for customer category
5 category = input("Enter the customer category (residential/commercial/industrial): ").lower()
6 # Calculate electricity usage
7 usage = new_reading - old_reading
8 # Initialize total cost variable
9 total_cost = 0
10 # Apply different calculation rules based on customer category
11 if category == "residential":
12     if usage <= 100:
13         total_cost = usage * 0.10
14     elif usage <= 300:
15         total_cost = (100 * 0.10) + ((usage - 100) * 0.15)
16     else:
17         total_cost = (100 * 0.10) + (200 * 0.15) + ((usage - 300) * 0.20)
18 elif category == "commercial":
19     if usage <= 200:
20         total_cost = usage * 0.12
21     elif usage <= 500:
22         total_cost = (200 * 0.12) + ((usage - 200) * 0.18)
23     else:
24         total_cost = (200 * 0.12) + (300 * 0.18) + ((usage - 500) * 0.25)
25 elif category == "industrial":
26     if usage <= 500:
27         total_cost = usage * 0.15
28     elif usage <= 1000:
29         total_cost = (500 * 0.15) + ((usage - 500) * 0.22)
30     else:
31         total_cost = (500 * 0.15) + (500 * 0.22) + ((usage - 1000) * 0.30)
32 else:
33     print("Invalid customer category.")
34 # Display the total electricity usage and cost
35 if total_cost > 0:
36     print(f"Total electricity usage: {usage} kWh")
37     print(f"Total cost: ${total_cost:.2f}")
```

Output :



```
PS C:\Users\devis\OneDrive\Desktop\AI Assisted Coding> & C:/Users/devis/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/devis/OneDrive/Desktop/AI Assisted Coding/assignment 3.3.py"
Enter the old meter reading: 25205
Enter the new meter reading: 32154
Enter the customer category (residential/commercial/industrial): industrial
Total electricity usage: 6949.0 kWh
Total cost: $1969.70
PS C:\Users\devis\OneDrive\Desktop\AI Assisted Coding>
```

Explanation :

- Takes old and new meter readings as user input.
- Reads the customer category (residential, commercial, or industrial).

- Calculates electricity units consumed by subtracting old reading from new reading.
- Applies different billing rules based on the customer category.
- Uses slab-wise rates to calculate total cost.
- Displays total units consumed and the final electricity bill amount

TASK – 02

Prompt : Optimize the code by removing redundant calculations and improving readability.

Code :

```
assignment 3.3.py •
assignment 3.3.py > ...
1  #Optimize the code by removing redundant calculations and improving readability.
2  old_reading = float(input("Enter the old meter reading: "))
3  new_reading = float(input("Enter the new meter reading: "))
4  category = input("Enter the customer category (residential/commercial/industrial): ").lower()
5  usage = new_reading - old_reading
6  total_cost = 0
7  if category == "residential":
8      if usage <= 100:
9          rate = 0.10
10         elif usage <= 300:
11             total_cost = (100 * 0.10) + ((usage - 100) * 0.15)
12         else:
13             total_cost = (100 * 0.10) + (200 * 0.15) + ((usage - 300) * 0.20)
14         if total_cost == 0:
15             total_cost = usage * rate
16     elif category == "commercial":
17         if usage <= 200:
18             rate = 0.12
19         elif usage <= 500:
20             total_cost = (200 * 0.12) + ((usage - 200) * 0.18)
21         else:
22             total_cost = (200 * 0.12) + (300 * 0.18) + ((usage - 500) * 0.25)
23         if total_cost == 0:
24             total_cost = usage * rate
25     elif category == "industrial":
26         if usage <= 500:
27             rate = 0.15
28         elif usage <= 1000:
29             total_cost = (500 * 0.15) + ((usage - 500) * 0.22)
30         else:
31             total_cost = (500 * 0.15) + (500 * 0.22) + ((usage - 1000) * 0.30)
32         if total_cost == 0:
33             total_cost = usage * rate
34     else:
35         print("Invalid customer category.")
36     if total_cost > 0:
37         print(f"Total electricity usage: {usage} kWh")
38         print(f"Total cost: ${total_cost:.2f}")
39
```

Output :

```

PS C:\Users\devis\OneDrive\Desktop\AI Assisted Coding> & C:/Users/devis/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/devis/OneDrive/Desktop/AI Assisted Coding/assignment 3.3.py"
Enter the old meter reading: 25205
Enter the new meter reading: 31265
Enter the customer category (residential/commercial/industrial): commercial
Total electricity usage: 6060.0 kWh
Total cost: $1468.00
PS C:\Users\devis\OneDrive\Desktop\AI Assisted Coding>

```

Explanation :

- Reads old and new meter readings and customer category from the user.
- Calculates units consumed by subtracting old reading from new reading.
- Uses conditional statements to apply different slab-wise rates for residential, commercial, and industrial users.
- Introduces a rate variable to avoid repeating calculations for the first slab.
- Checks if total_cost is zero and applies the first-slab rate if needed, reducing redundant computations.
- Handles invalid customer categories with a clear message.
- Displays total electricity usage and total cost in a readable format.

TASK – 03

Prompt : Generate a Python program using modular design (user-defined functions) for an electricity billing system. Uses user-defined functions to Calculate Energy Charges, Calculate Fixed Charges. Returns calculated values. Displays the total bill amount and Energy Charges and Fixed Charges separately. Use test cases to validate the program.

Code :

```

assignment 3.3.py X
assignment 3.3.py > ...
1 # Generate a Python program using modular design (user-defined functions) for an electricity billing system. Uses user
2 def calculate_energy_charges(units_consumed):
3     # Assuming a simple tariff structure for demonstration
4     if units_consumed <= 100:
5         return units_consumed * 5 # Rs. 5 per unit for first 100 units
6     elif units_consumed <= 300:
7         return (100 * 5) + ((units_consumed - 100) * 7) # Rs. 7 per unit for next 200 units
8     else:
9         return (100 * 5) + (200 * 7) + ((units_consumed - 300) * 10) # Rs. 10 per unit for above 300 units
10 def calculate_fixed_charges():
11     return 50 # Fixed charge of Rs. 50
12 def calculate_total_bill(units_consumed):
13     energy_charges = calculate_energy_charges(units_consumed)
14     fixed_charges = calculate_fixed_charges()
15     total_bill = energy_charges + fixed_charges
16     return energy_charges, fixed_charges, total_bill
17 def display_bill(units_consumed):
18     energy_charges, fixed_charges, total_bill = calculate_total_bill(units_consumed)
19     print(f"Units Consumed: {units_consumed}")
20     print(f"Energy Charges: Rs. {energy_charges}")
21     print(f"Fixed Charges: Rs. {fixed_charges}")
22     print(f"Total Bill Amount: Rs. {total_bill}")
23 # Test cases to validate the program
24 def test_billing_system():
25     test_cases = [50, 150, 350, 0, 1000]
26     for units in test_cases:
27         print("\n--- Bill Details ---")
28         display_bill(units)
29 test_billing_system()

```

Output :

```
PS C:\Users\devis\OneDrive\Desktop\AI Assisted Coding> & C:/Users/devis/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/devis/OneDrive/Desktop/AI Assisted Coding/assignment 3.3.py"
Units Consumed: 50
Energy Charges: Rs. 250
Fixed Charges: Rs. 50
Total Bill Amount: Rs. 300

--- Bill Details ---
Units Consumed: 150
Energy Charges: Rs. 850
Fixed Charges: Rs. 50
Total Bill Amount: Rs. 900

--- Bill Details ---
Units Consumed: 350
Energy Charges: Rs. 2400
Fixed Charges: Rs. 50
Total Bill Amount: Rs. 2450

--- Bill Details ---
Units Consumed: 0
Energy Charges: Rs. 0
Fixed Charges: Rs. 50
Total Bill Amount: Rs. 50

--- Bill Details ---
Units Consumed: 1000
Energy Charges: Rs. 8900
Fixed Charges: Rs. 50
Total Bill Amount: Rs. 8950
PS C:\Users\devis\OneDrive\Desktop\AI Assisted Coding>
```

Explanation :

- The program is designed using a modular approach with user-defined functions.
- `calculate_energy_charges()` computes energy charges based on slab-wise tariff rates.
- `calculate_fixed_charges()` returns a constant fixed charge.
- `calculate_total_bill()` calculates the total bill by adding energy charges and fixed charges.
- `display_bill()` prints units consumed, energy charges, fixed charges, and total bill separately.
- `test_billing_system()` uses multiple test cases to validate the correctness of the program.
- This modular design improves readability, reusability, and easy testing of the program.

TASK – 04

Prompt : Enhance the electricity billing program to include Fixed Charges (FC), Customer Charges (CC), and Electricity Duty (ED). Calculate ED as a percentage of Energy Charges (EC). Print individual values of EC, FC, CC, and ED, verify intermediate calculations, and display a clearly formatted final electricity bill showing the total payable amount.

Code :

```

1 # Enhance the electricity billing program to include Fixed Charges (FC), Customer Charges (CC), and Electricity Duty (ED)
2 # Calculate ED as a percentage of Energy Charges (EC).
3 # Print individual values of EC, FC, CC, and ED, verify intermediate calculations, and display a clearly formatted final bill.
4 # Constants
5 FC = 50.0 # Fixed Charges
6 CC = 30.0 # Customer Charges
7 ED_RATE = 0.05 # Electricity Duty rate (5%)
8 RATE_PER_UNIT = 0.12 # Rate per unit of electricity
9 # Function to calculate electricity bill
10 def calculate_electricity_bill(units_consumed):
11     # Calculate Energy Charges
12     EC = units_consumed * RATE_PER_UNIT
13
14     # Calculate Electricity Duty
15     ED = EC * ED_RATE
16
17     # Calculate Total Payable Amount
18     total_payable = EC + FC + CC + ED
19
20     # Print individual components
21     print(f"Energy Charges (EC): ${EC:.2f}")
22     print(f"Fixed Charges (FC): ${FC:.2f}")
23     print(f"Customer Charges (CC): ${CC:.2f}")
24     print(f"Electricity Duty (ED): ${ED:.2f}")
25
26     # Print final bill
27     print("\n----- Electricity Bill -----")
28     print(f"Total Payable Amount: ${total_payable:.2f}")
29     print("-----")
30
31     return total_payable
32
33 # Example usage
34 units = 150 # Example units consumed
35 calculate_electricity_bill(units)

```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] ...
```

```
PS C:\Users\devis\OneDrive\Desktop\AI Assisted Coding> & c:/Users/devis/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/devis/OneDrive/Desktop/AI Assisted Coding/assignment 3.3.py"
```

```
Energy Charges (EC): $18.00  
Fixed Charges (FC): $50.00  
Customer Charges (CC): $30.00  
Electricity Duty (ED): $0.90  
  
----- Electricity Bill -----  
Total Payable Amount: $98.90  
-----
```

```
PS C:\Users\devis\OneDrive\Desktop\AI Assisted Coding>
```

Explanation :

- The program calculates the electricity bill based on units consumed.
- Energy Charges (EC) are computed by multiplying units consumed with the rate per unit.
- Fixed Charges (FC) and Customer Charges (CC) are constant values added to the bill.
- Electricity Duty (ED) is calculated as a fixed percentage (5%) of the energy charges.

- All individual charges (EC, FC, CC, ED) are printed separately for verification.
- The total payable amount is obtained by adding EC, FC, CC, and ED.
- The final bill is displayed in a clear and structured format.

TASK – 05

Prompt : Develop a Python program to calculate an electricity bill. Compute Energy Charges (EC) based on units consumed, add Fixed Charges (FC) and Customer Charges (CC), and calculate Electricity Duty (ED) as a percentage of EC. Calculate the Total Bill using the formula: $\text{Total Bill} = \text{EC} + \text{FC} + \text{CC} + \text{ED}$. Display Energy Charges (EC), Fixed Charges (FC), Customer Charges (CC), Electricity Duty (ED), and the final Total Bill Amount in a clear and wellformatted output.

Code :

```

assignment 3.3.py > ...
1  # Develop a Python program to calculate an electricity bill.
2  #Compute Energy Charges (EC) based on units consumed, add Fixed Charges (FC) and Customer Charges (CC), and calculate
3  #Calculate the Total Bill using the formula:
4  #Total Bill = EC + FC + CC + ED.
5  #Display Energy Charges (EC), Fixed Charges (FC), Customer Charges (CC), Electricity Duty (ED), and the final Total Bi
6  def calculate_electricity_bill(units_consumed):
7      # Define rates and charges
8      rate_per_unit = 5.0 # Example rate per unit in currency
9      fixed_charges = 50.0 # Fixed charges in currency
10     customer_charges = 20.0 # Customer charges in currency
11     electricity_duty_rate = 0.05 # Electricity duty rate (5%)
12
13     # Calculate Energy Charges (EC)
14     energy_charges = units_consumed * rate_per_unit
15
16     # Calculate Electricity Duty (ED)
17     electricity_duty = energy_charges * electricity_duty_rate
18
19     # Calculate Total Bill
20     total_bill = energy_charges + fixed_charges + customer_charges + electricity_duty
21
22     # Display the breakdown of charges
23     print(f"Energy Charges (EC): {energy_charges:.2f}")
24     print(f"Fixed Charges (FC): {fixed_charges:.2f}")
25     print(f"Customer Charges (CC): {customer_charges:.2f}")
26     print(f"Electricity Duty (ED): {electricity_duty:.2f}")
27     print(f"Total Bill Amount: {total_bill:.2f}")
28
29 # Example usage
30 units = float(input("Enter the number of units consumed: "))
31 calculate_electricity_bill(units)

```

Output :



```
PS C:\Users\devis\OneDrive\Desktop\AI Assisted Coding> C:/Users/devis/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/devis/OneDrive/Desktop/AI Assisted Coding/assignment 3.3.py"
Enter the number of units consumed: 56.5
Energy Charges (EC): 282.50
Fixed Charges (FC): 50.00
Customer Charges (CC): 20.00
Electricity Duty (ED): 14.12
Total Bill Amount: 366.62
PS C:\Users\devis\OneDrive\Desktop\AI Assisted Coding>
```

Explanation :

- The program calculates the electricity bill based on the number of units consumed.
- A function `calculate_electricity_bill()` is used to perform all calculations.
- Rate per unit, Fixed Charges (FC), Customer Charges (CC), and Electricity Duty rate are defined inside the function.
- Energy Charges (EC) are calculated by multiplying units consumed with the rate per unit.
- Electricity Duty (ED) is calculated as 5% of the Energy Charges.
- The Total Bill Amount is calculated using the formula: $\text{Total Bill} = \text{EC} + \text{FC} + \text{CC} + \text{ED}$.
- All individual charges (EC, FC, CC, ED) are printed clearly.
- Finally, the total payable amount is displayed in a well-formatted manner.