ASSIGNMENT 3.3

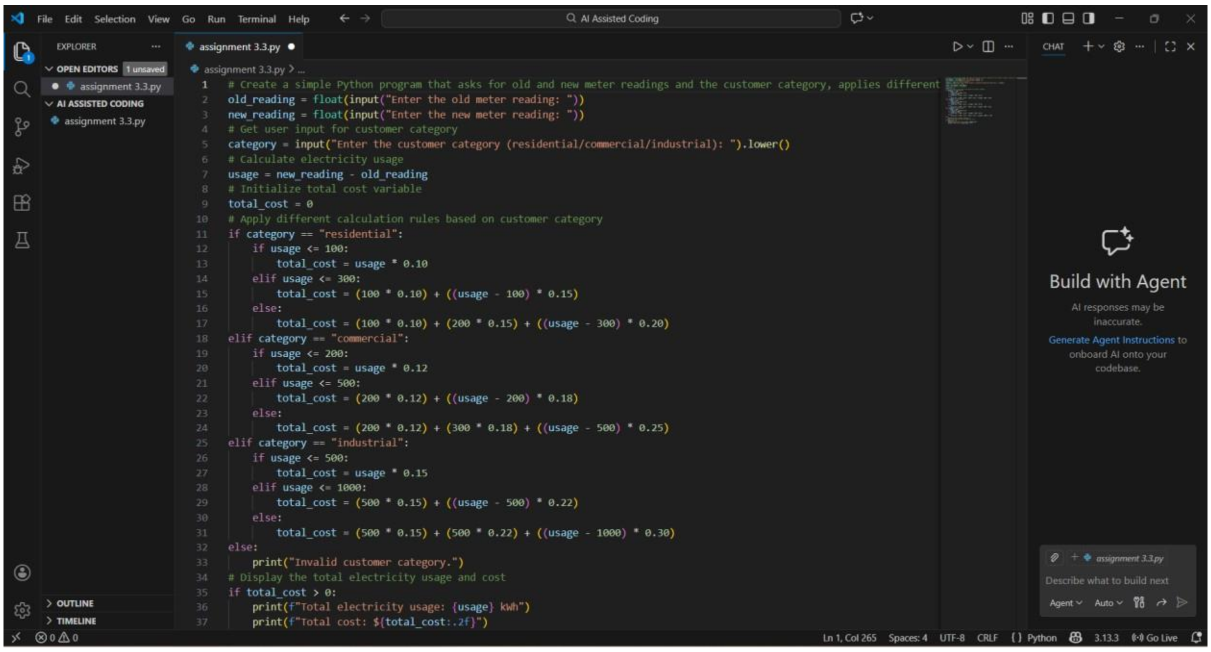
Ht.No : 2303A51584

Batch – 05

Task – 01

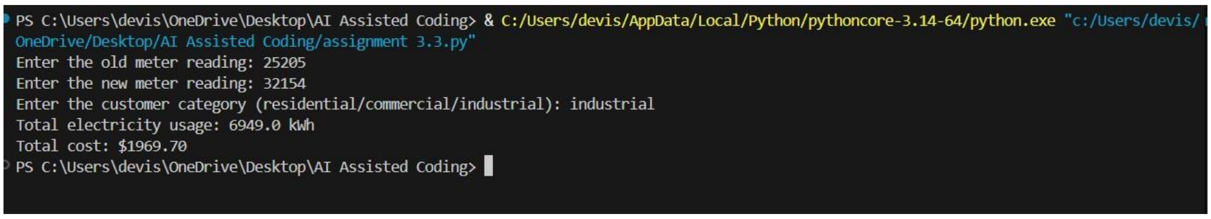
Prompt : Create a simple Python program that asks for old and new meter readings and the customer category, applies different calculation rules for each category, and displays the total electricity usage. Write all logic directly in the main code without using functions.

Code :



```python
# Create a simple Python program that asks for old and new meter readings and the customer category, applies different
old_reading = float(input("Enter the old meter reading: "))
new_reading = float(input("Enter the new meter reading: "))
# Get user input for customer category
category = input("Enter the customer category (residential/commercial/industrial): ").lower()
# Calculate electricity usage
usage = new_reading - old_reading
# Initialize total cost variable
total_cost = 0
# Apply different calculation rules based on customer category
if category == "residential":
    if usage <= 100:
        total_cost = usage * 0.10
    elif usage <= 300:
        total_cost = (100 * 0.10) + ((usage - 100) * 0.15)
    else:
        total_cost = (100 * 0.10) + (200 * 0.15) + ((usage - 300) * 0.20)
elif category == "commercial":
    if usage <= 200:
        total_cost = usage * 0.12
    elif usage <= 500:
        total_cost = (200 * 0.12) + ((usage - 200) * 0.18)
    else:
        total_cost = (200 * 0.12) + (300 * 0.18) + ((usage - 500) * 0.25)
elif category == "industrial":
    if usage <= 500:
        total_cost = usage * 0.15
    elif usage <= 1000:
        total_cost = (500 * 0.15) + ((usage - 500) * 0.22)
    else:
        total_cost = (500 * 0.15) + (500 * 0.22) + ((usage - 1000) * 0.30)
else:
    print("Invalid customer category.")
# Display the total electricity usage and cost
if total_cost > 0:
    print(f"Total electricity usage: {usage} kWh")
    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>
```

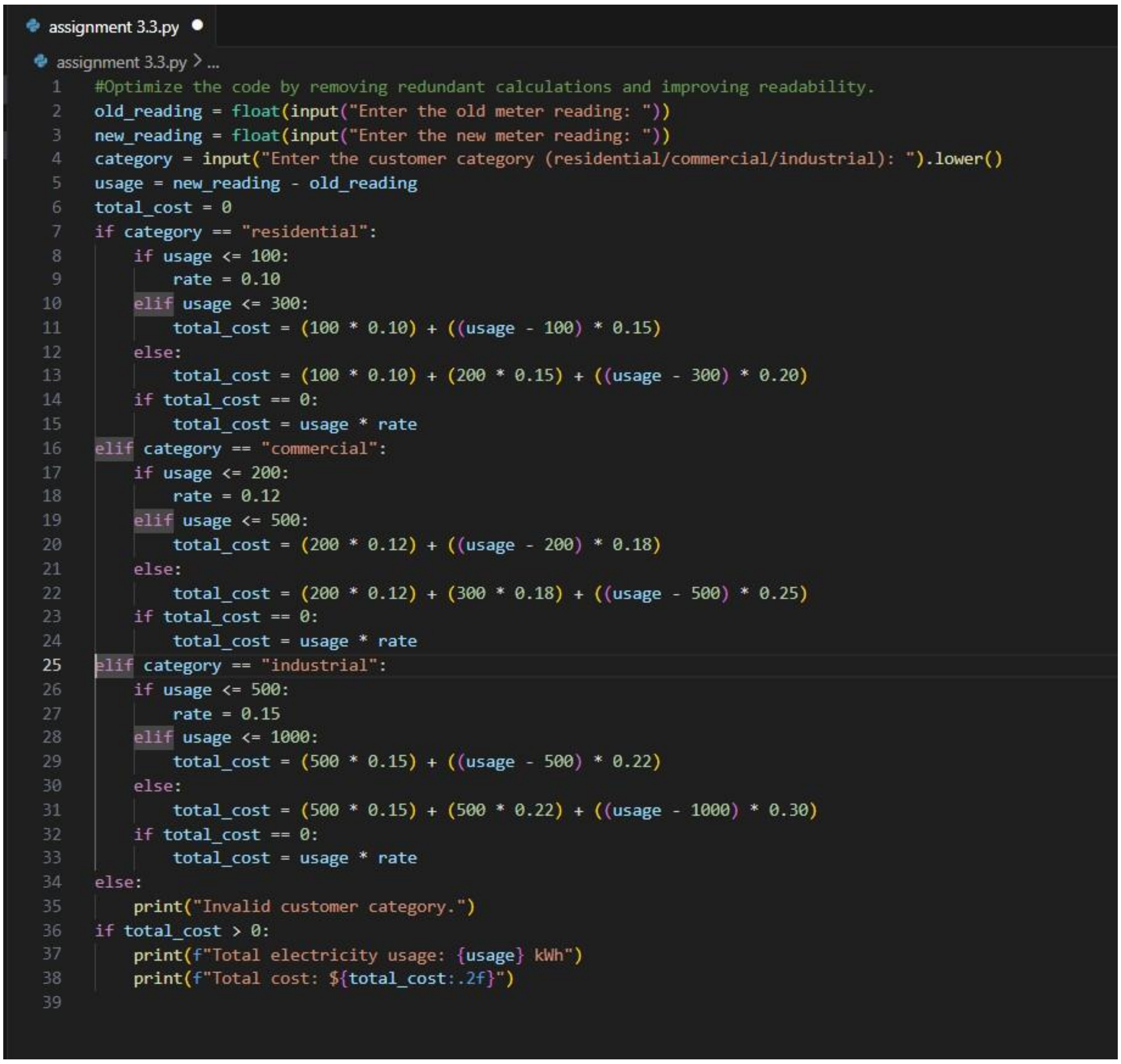
Explanation :

- 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

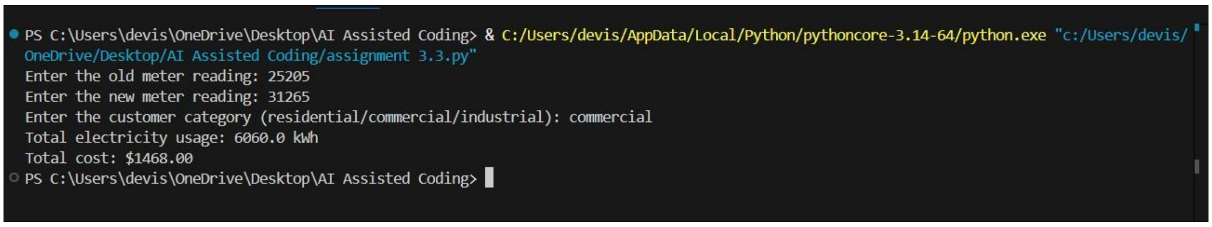
Prompt : Optimize the code by removing redundant calculations and improving readability.

Code :



```python
#Optimize the code by removing redundant calculations and improving readability.
old_reading = float(input("Enter the old meter reading: "))
new_reading = float(input("Enter the new meter reading: "))
category = input("Enter the customer category (residential/commercial/industrial): ").lower()
usage = new_reading - old_reading
total_cost = 0
if category == "residential":
    if usage <= 100:
        rate = 0.10
    elif usage <= 300:
        total_cost = (100 * 0.10) + ((usage - 100) * 0.15)
    else:
        total_cost = (100 * 0.10) + (200 * 0.15) + ((usage - 300) * 0.20)
    if total_cost == 0:
        total_cost = usage * rate
elif category == "commercial":
    if usage <= 200:
        rate = 0.12
    elif usage <= 500:
        total_cost = (200 * 0.12) + ((usage - 200) * 0.18)
    else:
        total_cost = (200 * 0.12) + (300 * 0.18) + ((usage - 500) * 0.25)
    if total_cost == 0:
        total_cost = usage * rate
elif category == "industrial":
    if usage <= 500:
        rate = 0.15
    elif usage <= 1000:
        total_cost = (500 * 0.15) + ((usage - 500) * 0.22)
    else:
        total_cost = (500 * 0.15) + (500 * 0.22) + ((usage - 1000) * 0.30)
    if total_cost == 0:
        total_cost = usage * rate
else:
    print("Invalid customer category.")
if total_cost > 0:
    print(f"Total electricity usage: {usage} kWh")
    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: 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 :

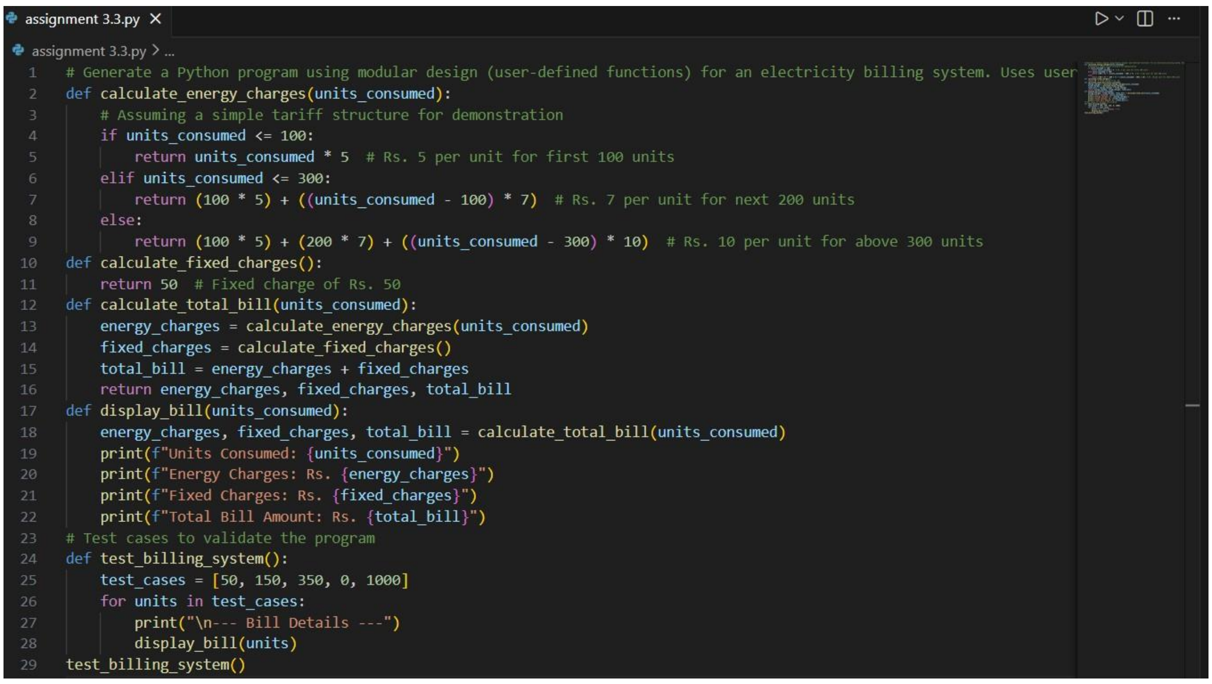
- Reads old and new meter readings and customer category from the user.
- Calculates units consumed by subtracting old reading from new reading.
- Uses conditional statements to apply different slab-wise rates for residential, commercial, and industrial users.
- Introduces a rate variable to avoid repeating 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

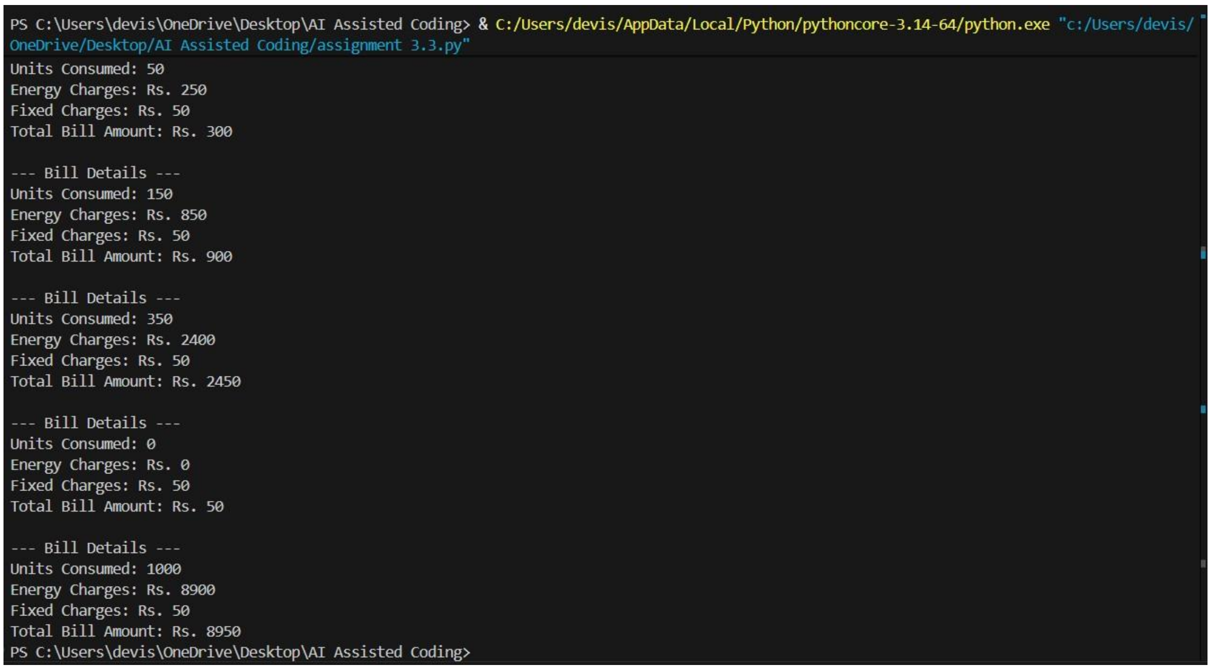
Calculate Energy Charges, Calculate Fixed Charges. Returns calculated values. Displays the total bill amount and Energy Charges and Fixed Charges separately**.** Use test cases to validate the program.

Code :



```python
# Generate a Python program using modular design (user-defined functions) for an electricity billing system. Uses user
def calculate_energy_charges(units_consumed):
    # Assuming a simple tariff structure for demonstration
    if units_consumed <= 100:
        return units_consumed * 5  # Rs. 5 per unit for first 100 units
    elif units_consumed <= 300:
        return (100 * 5) + ((units_consumed - 100) * 7)  # Rs. 7 per unit for next 200 units
    else:
        return (100 * 5) + (200 * 7) + ((units_consumed - 300) * 10)  # Rs. 10 per unit for above 300 units
def calculate_fixed_charges():
    return 50  # Fixed charge of Rs. 50
def calculate_total_bill(units_consumed):
    energy_charges = calculate_energy_charges(units_consumed)
    fixed_charges = calculate_fixed_charges()
    total_bill = energy_charges + fixed_charges
    return energy_charges, fixed_charges, total_bill
def display_bill(units_consumed):
    energy_charges, fixed_charges, total_bill = calculate_total_bill(units_consumed)
    print(f"Units Consumed: {units_consumed}")
    print(f"Energy Charges: Rs. {energy_charges}")
    print(f"Fixed Charges: Rs. {fixed_charges}")
    print(f"Total Bill Amount: Rs. {total_bill}")
# Test cases to validate the program
def test_billing_system():
    test_cases = [50, 150, 350, 0, 1000]
    for units in test_cases:
        print("\n--- Bill Details ---")
        display_bill(units)
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>
```

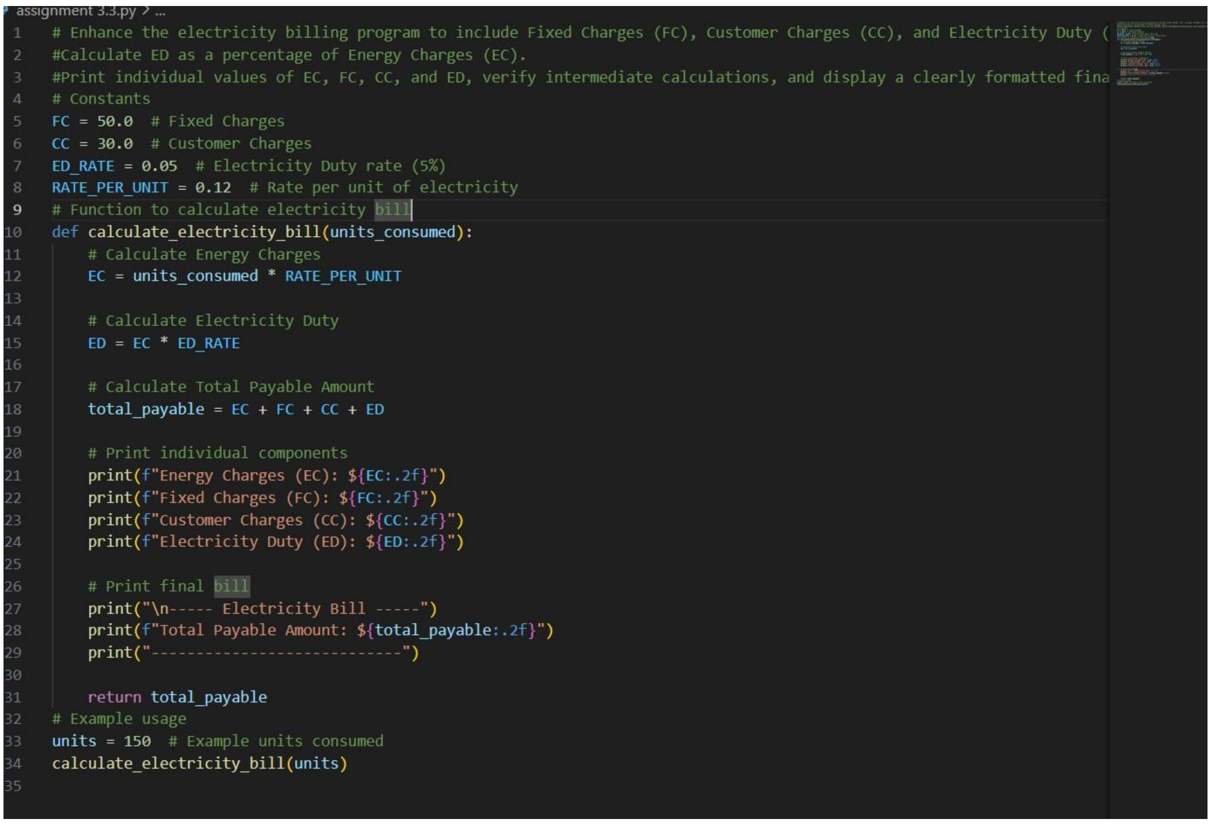
Explanation :

- 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, verifyintermediate calculations, and display a clearly formatted final electricity bill showing the total payable amount.

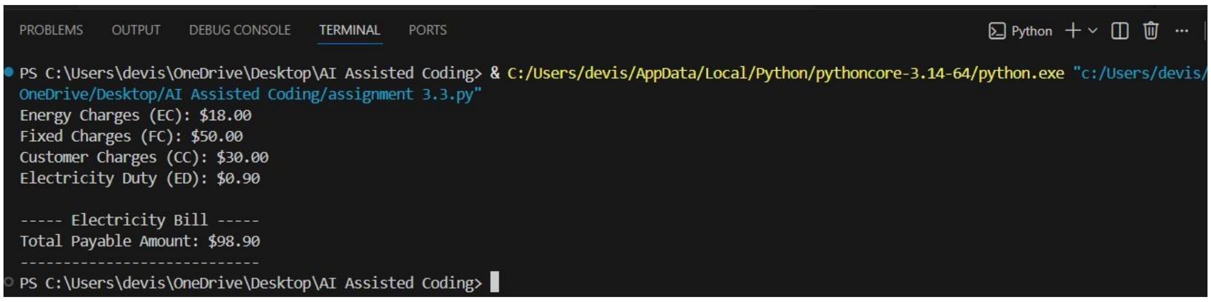
Code :



assignment 3.3.py > ...

```python
# Enhance the electricity billing program to include Fixed Charges (FC), Customer Charges (CC), and Electricity Duty (
#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 fina
# Constants
FC = 50.0  # Fixed Charges
CC = 30.0  # Customer Charges
ED_RATE = 0.05  # Electricity Duty rate (5%)
RATE_PER_UNIT = 0.12  # Rate per unit of electricity
# Function to calculate electricity bill
def calculate_electricity_bill(units_consumed):
    # Calculate Energy Charges
    EC = units_consumed * RATE_PER_UNIT

    # Calculate Electricity Duty
    ED = EC * ED_RATE

    # Calculate Total Payable Amount
    total_payable = EC + FC + CC + ED

    # Print individual components
    print(f"Energy Charges (EC): ${EC:.2f}")
    print(f"Fixed Charges (FC): ${FC:.2f}")
    print(f"Customer Charges (CC): ${CC:.2f}")
    print(f"Electricity Duty (ED): ${ED:.2f}")

    # Print final bill
    print("\n----- Electricity Bill -----")
    print(f"Total Payable Amount: ${total_payable:.2f}")
    print("----------------------------")

    return total_payable
# Example usage
units = 150  # Example units consumed
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"
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 :

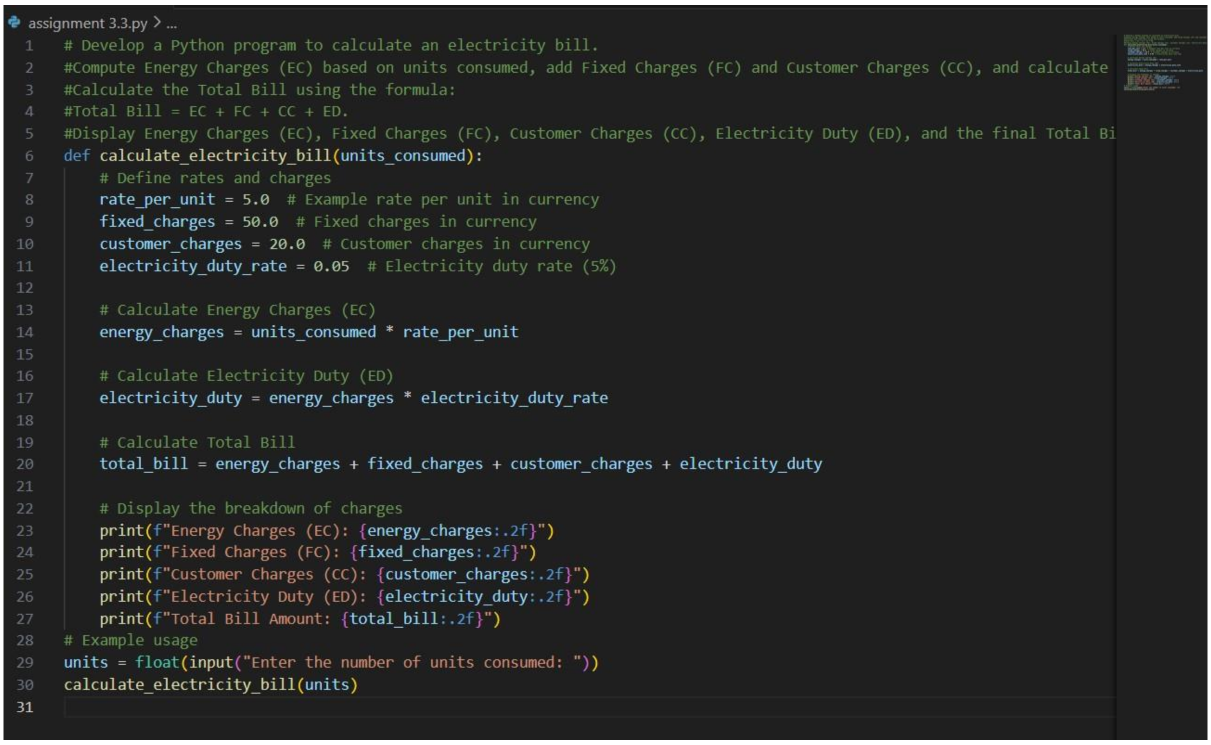
- The program calculates the electricity bill based on units consumed.
- Energy Charges (EC) are computed by multiplying units consumed with the rate per unit.
- Fixed Charges (FC) and Customer Charges (CC) are constant values added to the bill.
- Electricity Duty (ED) is calculated as a fixed percentage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

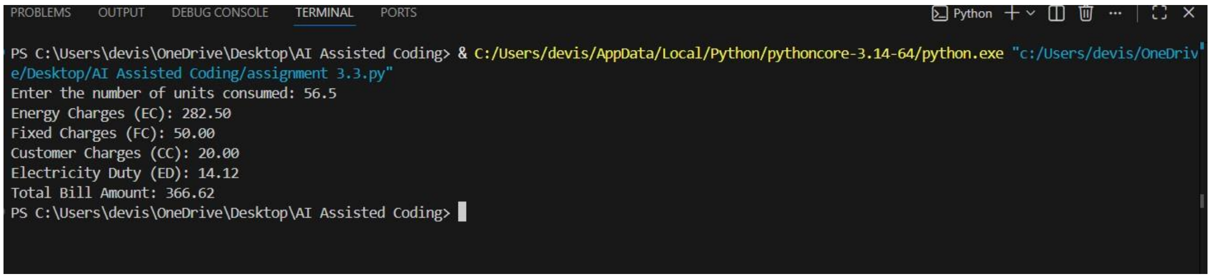
EC. Calculate the Total Bill using the formula: Total Bill = EC + FC + CC +
ED. Display Energy Charges (EC), Fixed Charges (FC), Customer Charges (CC), Electricity Duty (ED), and the final Total Bill Amount in a clear and wellformatted output.

Code :



```python
# 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
#Calculate the Total Bill using the formula:
#Total Bill = EC + FC + CC + ED.
#Display Energy Charges (EC), Fixed Charges (FC), Customer Charges (CC), Electricity Duty (ED), and the final Total Bi
def calculate_electricity_bill(units_consumed):
    # Define rates and charges
    rate_per_unit = 5.0  # Example rate per unit in currency
    fixed_charges = 50.0  # Fixed charges in currency
    customer_charges = 20.0  # Customer charges in currency
    electricity_duty_rate = 0.05  # Electricity duty rate (5%)

    # Calculate Energy Charges (EC)
    energy_charges = units_consumed * rate_per_unit

    # Calculate Electricity Duty (ED)
    electricity_duty = energy_charges * electricity_duty_rate

    # Calculate Total Bill
    total_bill = energy_charges + fixed_charges + customer_charges + electricity_duty

    # Display the breakdown of charges
    print(f"Energy Charges (EC): {energy_charges:.2f}")
    print(f"Fixed Charges (FC): {fixed_charges:.2f}")
    print(f"Customer Charges (CC): {customer_charges:.2f}")
    print(f"Electricity Duty (ED): {electricity_duty:.2f}")
    print(f"Total Bill Amount: {total_bill:.2f}")
# Example usage
units = float(input("Enter the number of units consumed: "))
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/OneDriv
e/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:Total Bill = EC + FC + CC + ED.
- All individual charges (EC, FC, CC, ED) are printed clearly.
- Finally, the total payable amount is displayed in a well-formatted manner.