Name: V.HRK SRI HASRHA VIGNESH

Batch:42

Hll.no:2303A52304

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

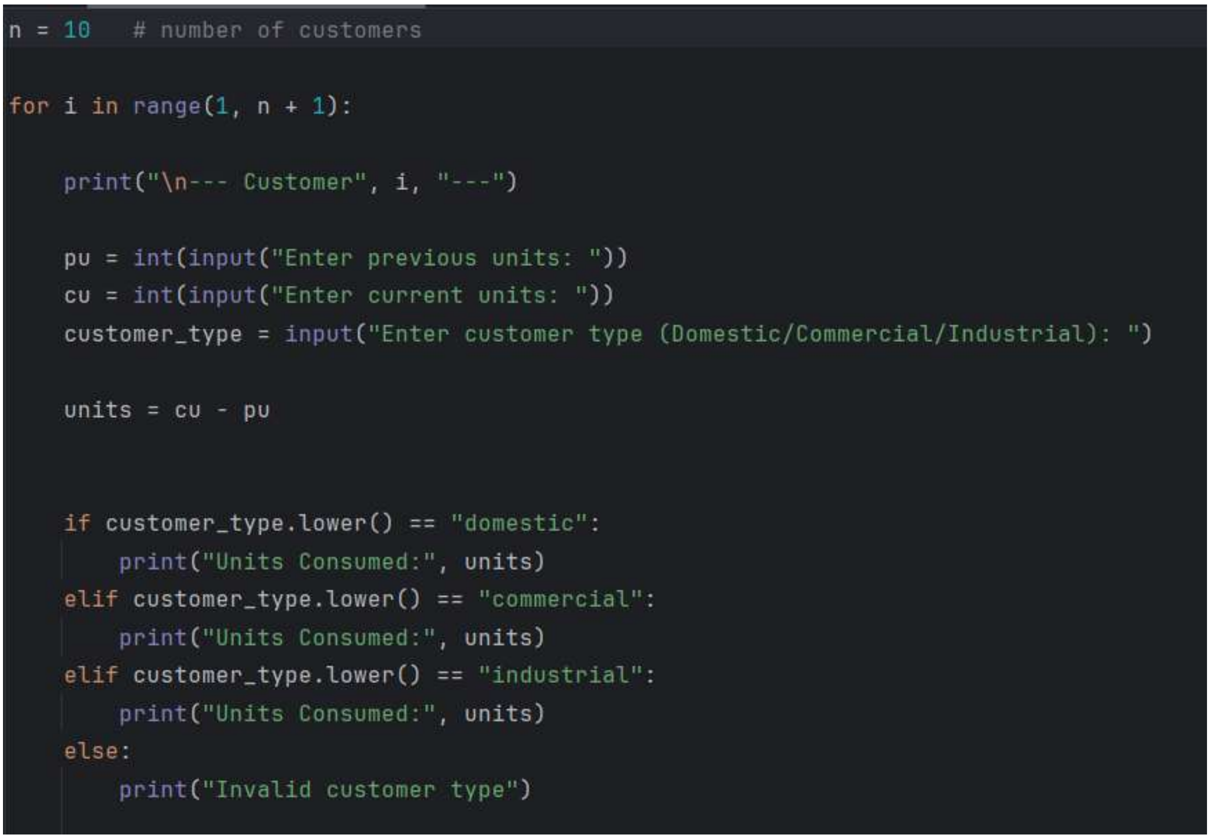
Task 1: Reading Consumer Details

Objective:

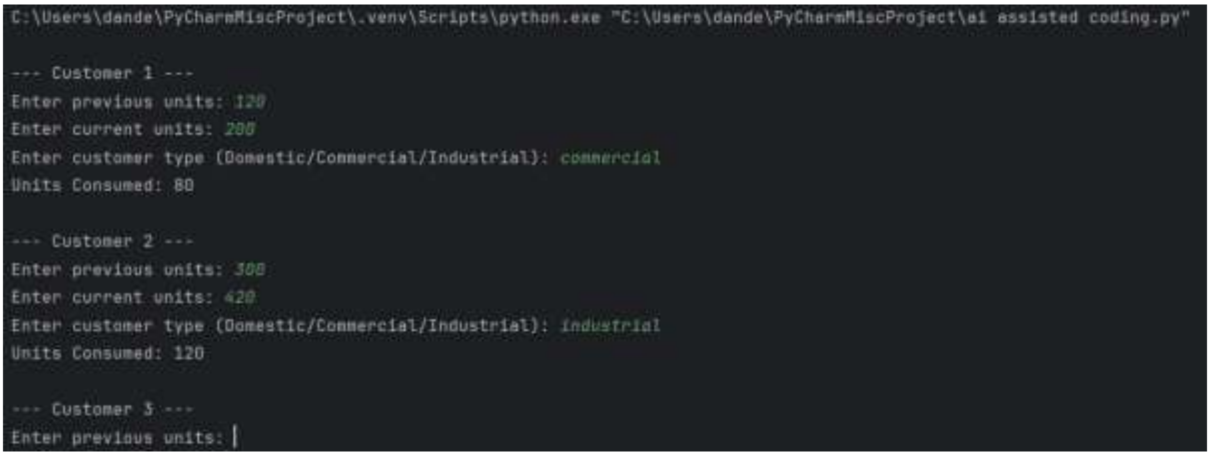
To read consumer details and calculate the units consumed.

AI Prompt Used:

"Generate Python code to read previous units, current units and calculate units consumed"



```python
n = 10   # number of customers

for i in range(1, n + 1):

    print("\n--- Customer", i, "---")

    pu = int(input("Enter previous units: "))
    cu = int(input("Enter current units: "))
    customer_type = input("Enter customer type (Domestic/Commercial/Industrial): ")

    units = cu - pu


    if customer_type.lower() == "domestic":
        print("Units Consumed:", units)
    elif customer_type.lower() == "commercial":
        print("Units Consumed:", units)
    elif customer_type.lower() == "industrial":
        print("Units Consumed:", units)
    else:
        print("Invalid customer type")
```



```
C:\Users\dande\PyCharmMiscProject\.venv\Scripts\python.exe "C:\Users\dande\PyCharmMiscProject\ai assisted coding.py"

--- Customer 1 ---
Enter previous units: 120
Enter current units: 200
Enter customer type (Domestic/Commercial/Industrial): commercial
Units Consumed: 80

--- Customer 2 ---
Enter previous units: 300
Enter current units: 420
Enter customer type (Domestic/Commercial/Industrial): industrial
Units Consumed: 120

--- Customer 3 ---
Enter previous units: 
```

Task 2: Energy Charges Calculation Based on Units Consumed

Objective

To calculate energy charges based on units consumed and customer type.

AI Prompt Used

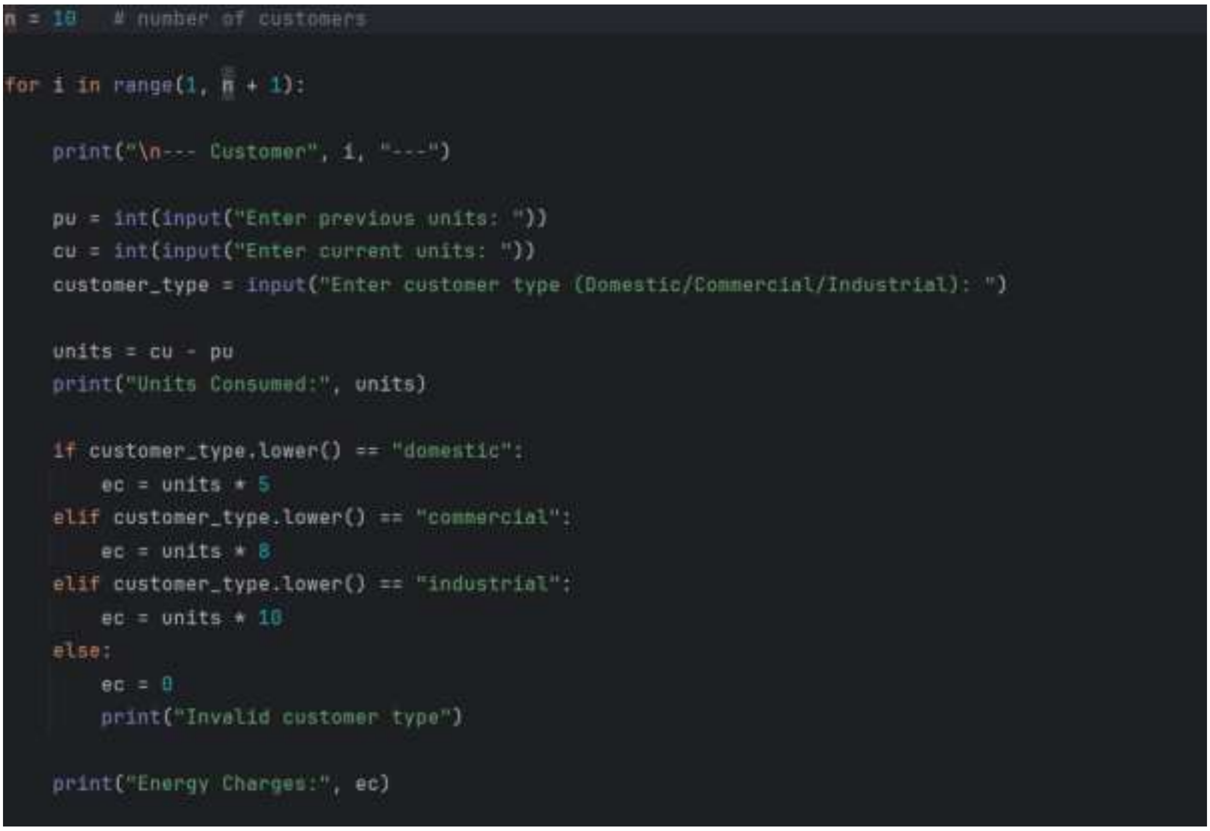
"Simplify energy charge calculation logic using if-else conditions"

Logic Explanation

Energy charges are calculated using conditional statements based on customer type:

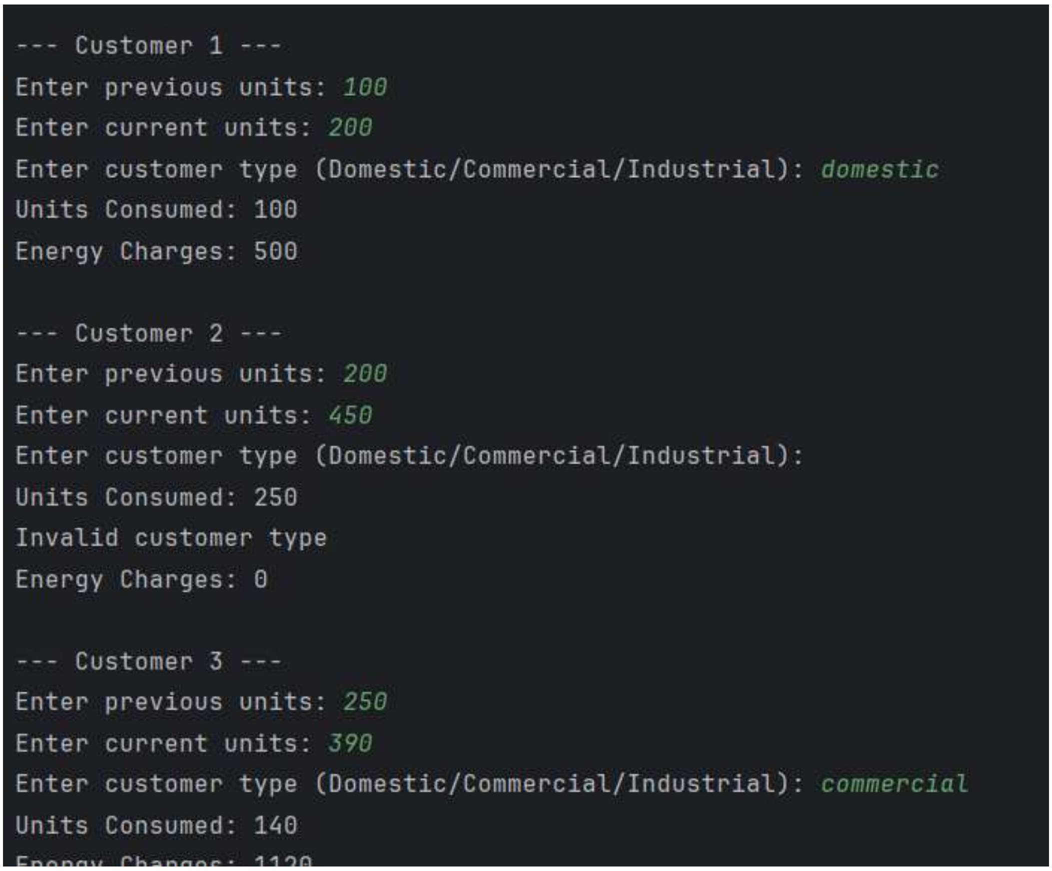
- Domestic: ₹5 per unit
- Commercial: ₹8 per unit
- Industrial: ₹10 per unit

Code:



```
n = 10   # number of customers

for i in range(1, n + 1):

    print("\n--- Customer", i, "---")

    pu = int(input("Enter previous units: "))
    cu = int(input("Enter current units: "))
    customer_type = input("Enter customer type (Domestic/Commercial/Industrial): ")

    units = cu - pu
    print("Units Consumed:", units)

    if customer_type.lower() == "domestic":
        ec = units * 5
    elif customer_type.lower() == "commercial":
        ec = units * 8
    elif customer_type.lower() == "industrial":
        ec = units * 10
    else:
        ec = 0
        print("Invalid customer type")

    print("Energy Charges:", ec)
```

Output:



```
--- Customer 1 ---
Enter previous units: 100
Enter current units: 200
Enter customer type (Domestic/Commercial/Industrial): domestic
Units Consumed: 100
Energy Charges: 500

--- Customer 2 ---
Enter previous units: 200
Enter current units: 450
Enter customer type (Domestic/Commercial/Industrial):
Units Consumed: 250
Invalid customer type
Energy Charges: 0

--- Customer 3 ---
Enter previous units: 250
Enter current units: 390
Enter customer type (Domestic/Commercial/Industrial): commercial
Units Consumed: 140
Energy Charges: 1120
```

Task 3: Modular Design Using Functions

Objective

To improve code readability and reusability using functions.

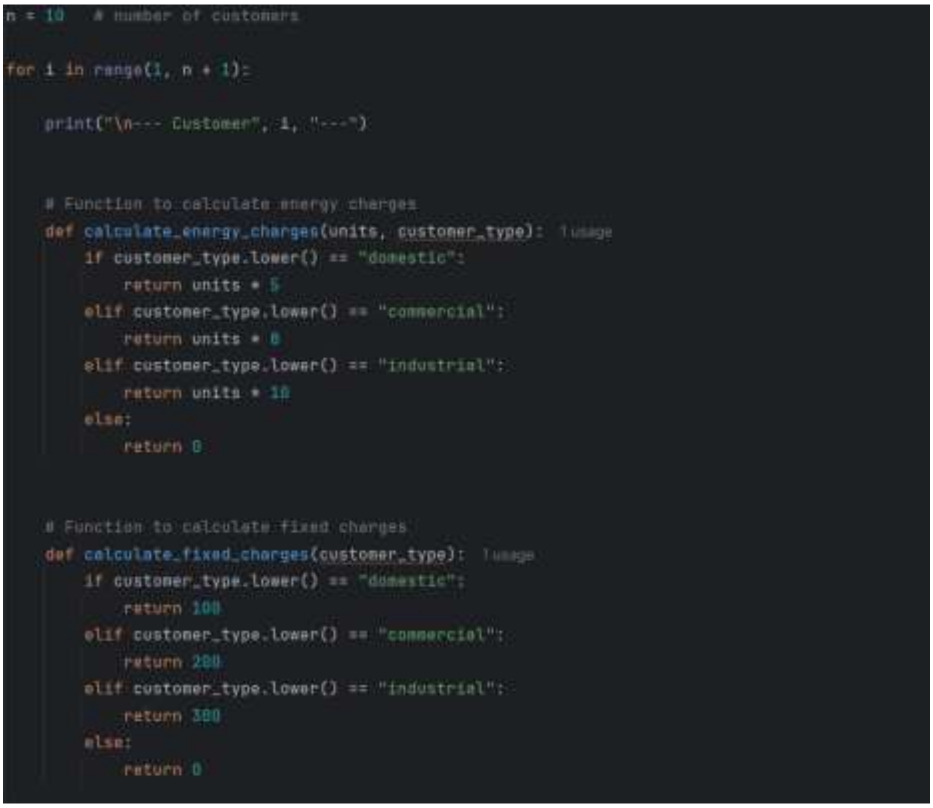
AI Prompt Used

"Create a Python function to calculate energy charges based on units and customer type"

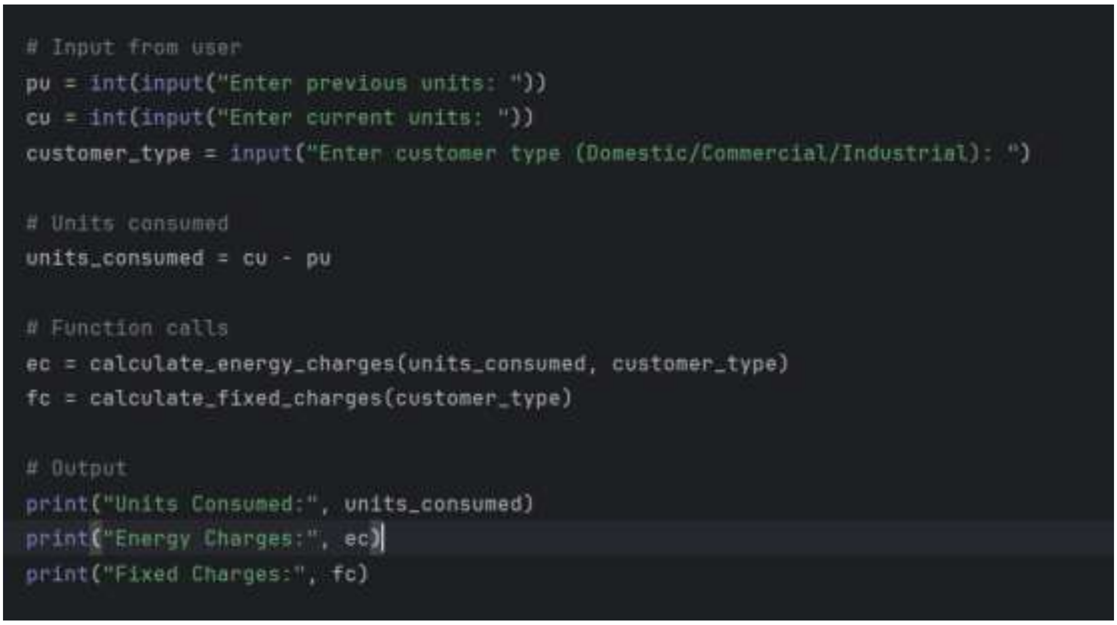
Logic Explanation

A user-defined function is used to calculate energy charges. This makes the program modular and easy to maintain.

Code:

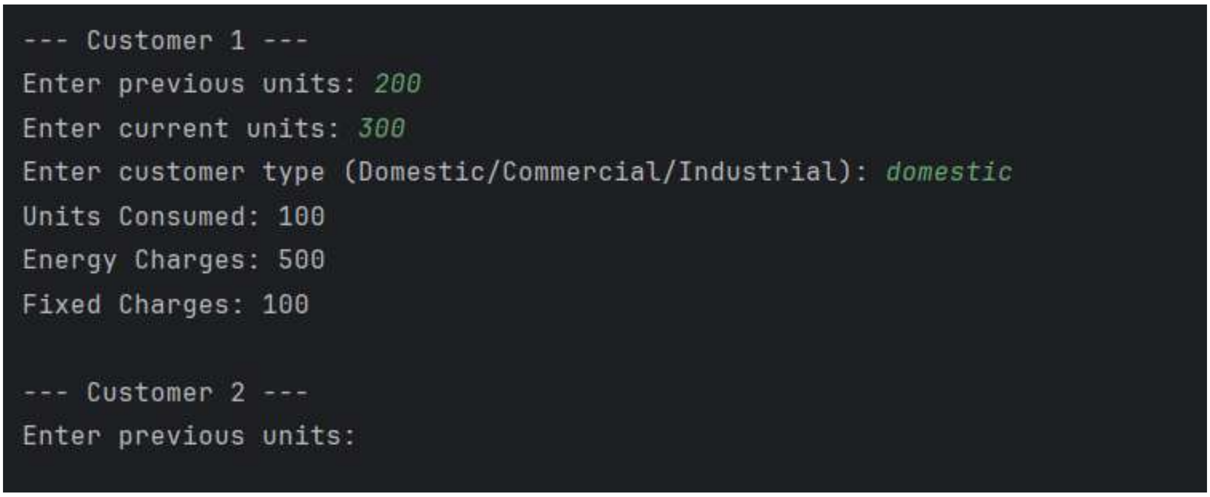


```python
n = 10   # number of customers

for i in range(1, n + 1):

    print("\n--- Customer", i, "---")


    # Function to calculate energy charges
    def calculate_energy_charges(units, customer_type):
        if customer_type.lower() == "domestic":
            return units * 5
        elif customer_type.lower() == "commercial":
            return units * 8
        elif customer_type.lower() == "industrial":
            return units * 10
        else:
            return 0


    # Function to calculate fixed charges
    def calculate_fixed_charges(customer_type):
        if customer_type.lower() == "domestic":
            return 100
        elif customer_type.lower() == "commercial":
            return 200
        elif customer_type.lower() == "industrial":
            return 300
        else:
            return 0
```



```
# Input from user
pu = int(input("Enter previous units: "))
cu = int(input("Enter current units: "))
customer_type = input(“Enter customer type (Domestic/Commercial/Industrial): ”)

# Units consumed
units_consumed = cu - pu

# Function calls
ec = calculate_energy_charges(units_consumed, customer_type)
fc = calculate_fixed_charges(customer_type)

# Output
print("Units Consumed:", units_consumed)
print("Energy Charges:", ec)
print("Fixed Charges:", fc)
```

Output:



```
--- Customer 1 ---
Enter previous units: 200
Enter current units: 300
Enter customer type (Domestic/Commercial/Industrial): domestic
Units Consumed: 100
Energy Charges: 500
Fixed Charges: 100

--- Customer 2 ---
Enter previous units:
```

Task 4: Calculation of Additional Charges

Objective

To calculate fixed charges, customer charges and electricity duty.

AI Prompt Used

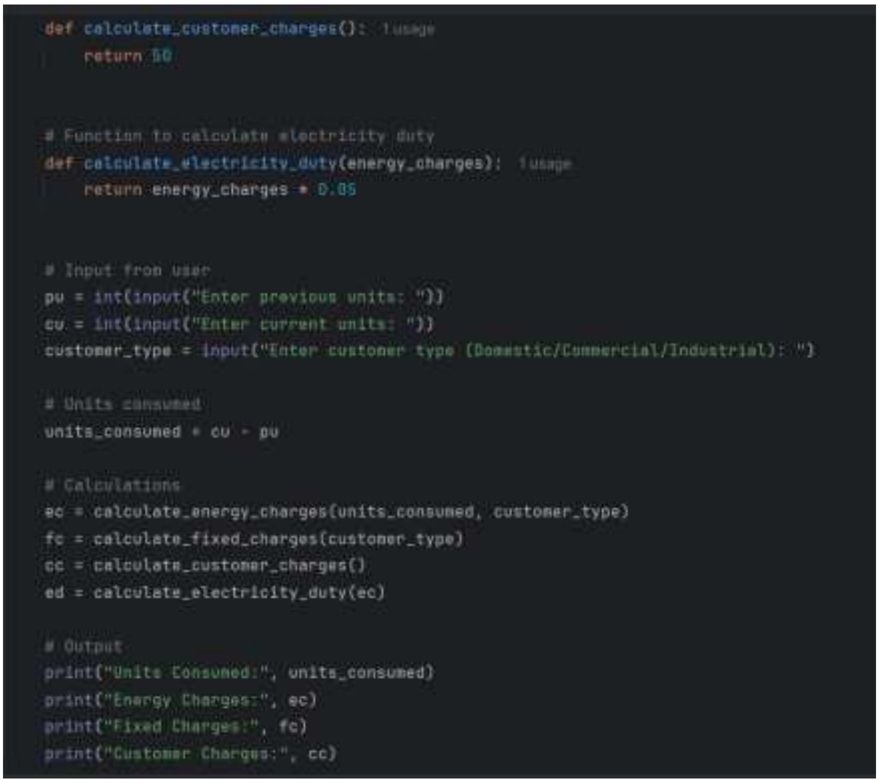
"Add electricity duty and fixed charges calculation to the program"

Logic Explanation

Additional charges include:

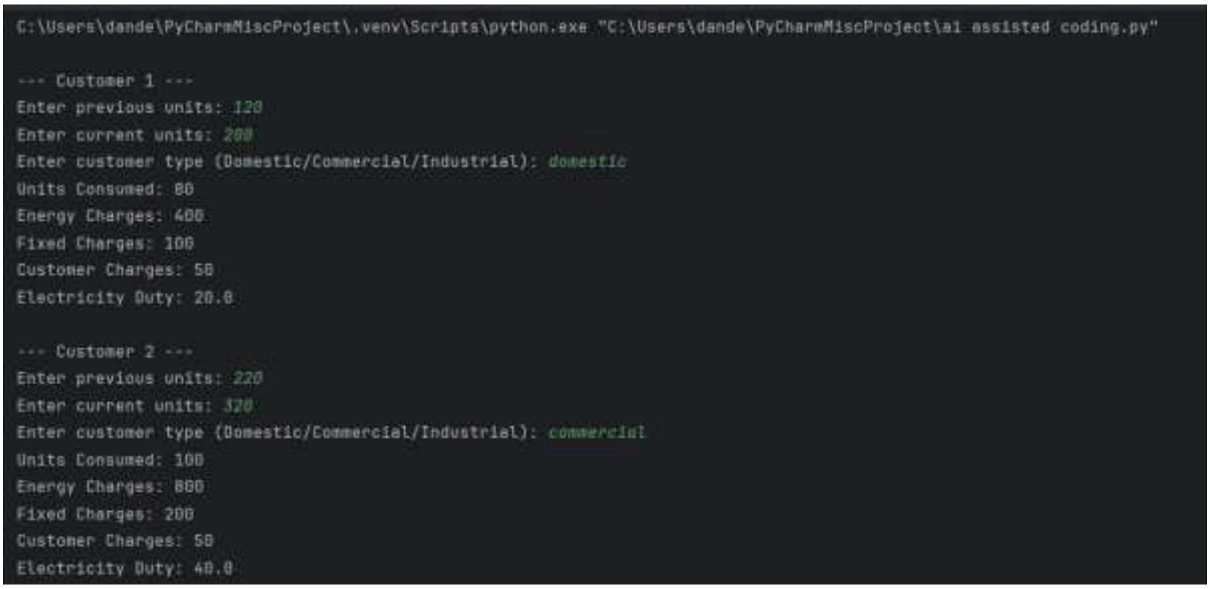
- Fixed Charges (FC): ₹100
- Customer Charges (CC): ₹50
- Electricity Duty (ED): 5% of Energy Charges

Code:



```
def calculate_customer_charges():  1 usage
    return 50


# Function to calculate electricity duty
def calculate_electricity_duty(energy_charges):  1 usage
    return energy_charges * 0.05


# Input from user
pu = int(input("Enter previous units: "))
cu = int(input("Enter current units: "))
customer_type = input("Enter customer type (Domestic/Commercial/Industrial): ")

# Units consumed
units_consumed = cu - pu

# Calculations
ec = calculate_energy_charges(units_consumed, customer_type)
fc = calculate_fixed_charges(customer_type)
cc = calculate_customer_charges()
ed = calculate_electricity_duty(ec)

# Output
print("Units Consumed:", units_consumed)
print("Energy Charges:", ec)
print("Fixed Charges:", fc)
print("Customer Charges:", cc)
```

Output:



```
C:\Users\dande\PyCharmMiscProject\.venv\Scripts\python.exe "C:\Users\dande\PyCharmMiscProject\ai assisted coding.py"

--- Customer 1 ---
Enter previous units: 120
Enter current units: 200
Enter customer type (Domestic/Commercial/Industrial): domestic
Units Consumed: 80
Energy Charges: 400
Fixed Charges: 100
Customer Charges: 50
Electricity Duty: 20.0

--- Customer 2 ---
Enter previous units: 220
Enter current units: 320
Enter customer type (Domestic/Commercial/Industrial): commercial
Units Consumed: 100
Energy Charges: 800
Fixed Charges: 200
Customer Charges: 50
Electricity Duty: 40.0
```

Task 5: Final Bill Generation

Objective

To calculate and display the final electricity bill.

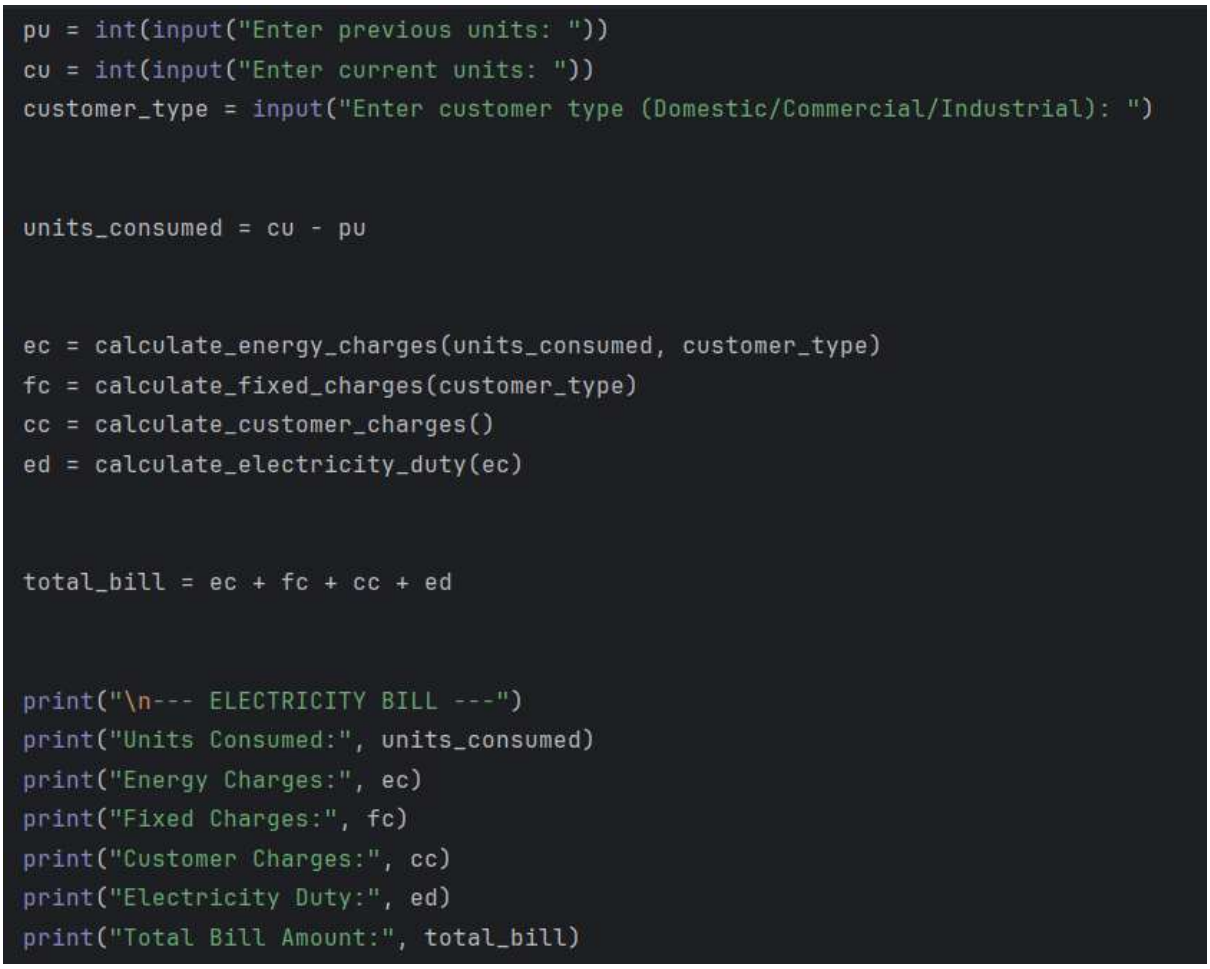
Logic Explanation

The total bill amount is calculated by adding all charges.

Formula

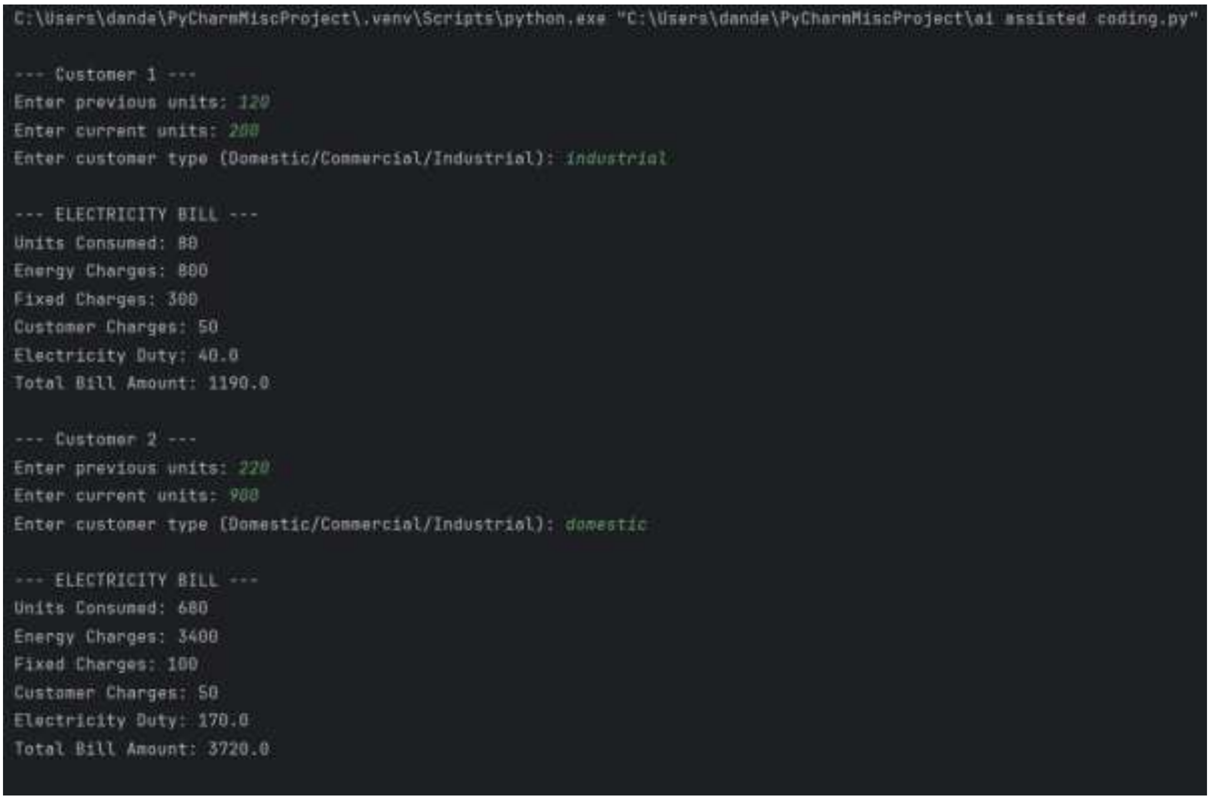
Total Bill = EC + FC + CC + ED

Code:



```python
pu = int(input("Enter previous units: "))
cu = int(input("Enter current units: "))
customer_type = input("Enter customer type (Domestic/Commercial/Industrial): ")


units_consumed = cu - pu


ec = calculate_energy_charges(units_consumed, customer_type)
fc = calculate_fixed_charges(customer_type)
cc = calculate_customer_charges()
ed = calculate_electricity_duty(ec)


total_bill = ec + fc + cc + ed


print("\n--- ELECTRICITY BILL ---")
print("Units Consumed:", units_consumed)
print("Energy Charges:", ec)
print("Fixed Charges:", fc)
print("Customer Charges:", cc)
print("Electricity Duty:", ed)
print("Total Bill Amount:", total_bill)
```

Output:



```
C:\Users\dande\PyCharmMiscProject\.venv\Scripts\python.exe "C:\Users\dande\PyCharmMiscProject\ai assisted coding.py"

--- Customer 1 ---
Enter previous units: 120
Enter current units: 200
Enter customer type (Domestic/Commercial/Industrial): industrial

--- ELECTRICITY BILL ---
Units Consumed: 80
Energy Charges: 800
Fixed Charges: 300
Customer Charges: 50
Electricity Duty: 40.0
Total Bill Amount: 1190.0

--- Customer 2 ---
Enter previous units: 220
Enter current units: 900
Enter customer type (Domestic/Commercial/Industrial): domestic

--- ELECTRICITY BILL ---
Units Consumed: 680
Energy Charges: 3400
Fixed Charges: 100
Customer Charges: 50
Electricity Duty: 170.0
Total Bill Amount: 3720.0
```

Analysis

The electricity billing application accurately calculates charges using conditional logic and modular programming. AI-assisted prompts helped in optimizing the logic and improving readability. The program closely resembles real-world electricity billing systems and is easy to extend for future enhancements.

Conclusion

This lab helped in understanding how Python and AI tools can be used to build real-world applications. The implementation of functions and conditional logic improved coding efficiency and clarity.