AI-ASSIST CODING LAB-3.3

SINDHU KATAKAM

BATCH-41

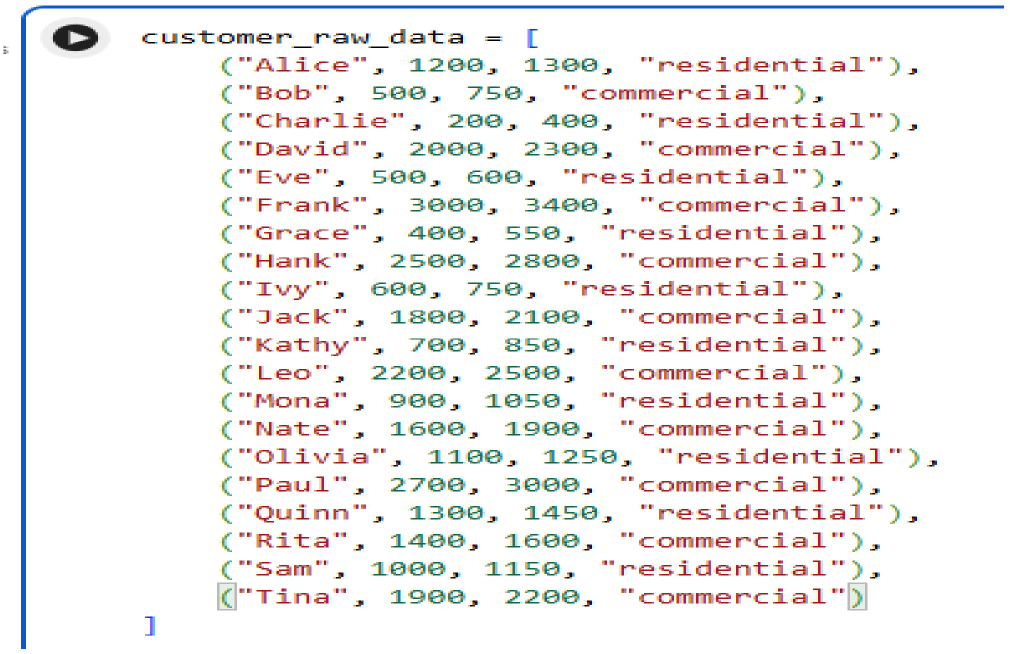
2303A52481

Task 1: AI-Generated Logic for Reading Consumer Details

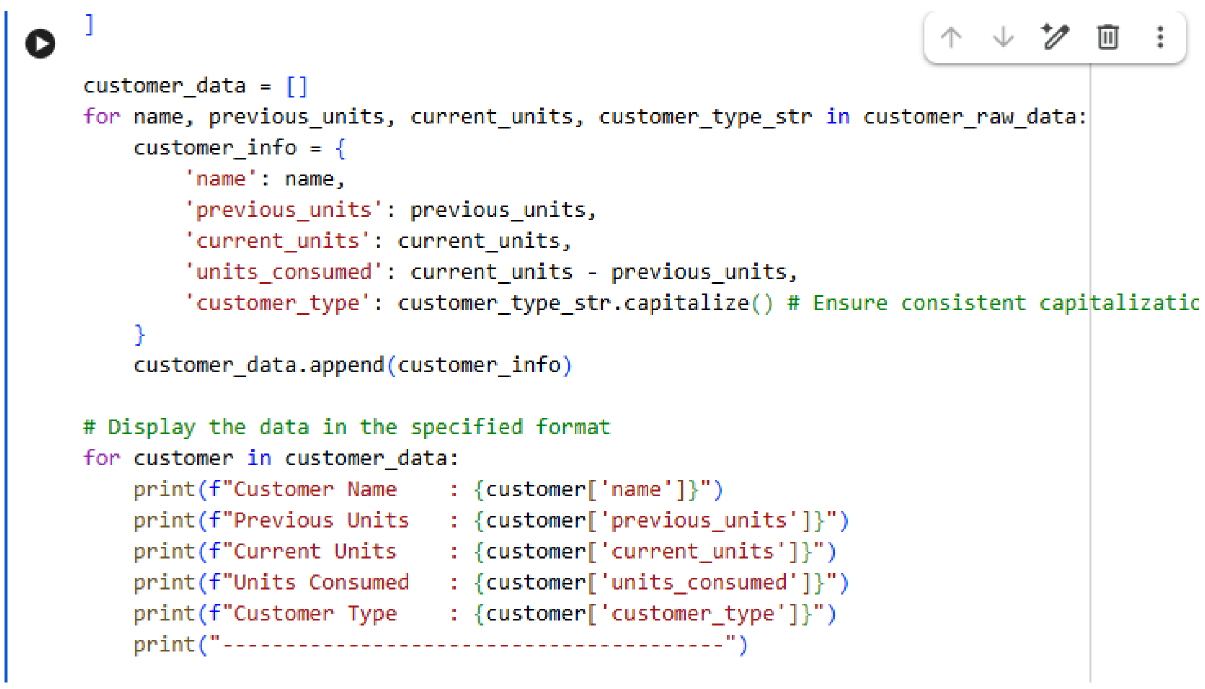
Prompt:

design a real-world electricity billing application using Python.it has to Apply conditional logic for tariff-based billing.now your task is to take data of 30 people randomly in the code itself that has to include name,units consumed,customer type (residential/commercial)and then Reads the Previous Units (PU), Current Units (CU), calculate the units consumed based on that data.give the output in this format customer name: sindhu, Previous Units : 120 Current Units : 180 Units Consumed : 60 Customer Type : Domestic

Code:

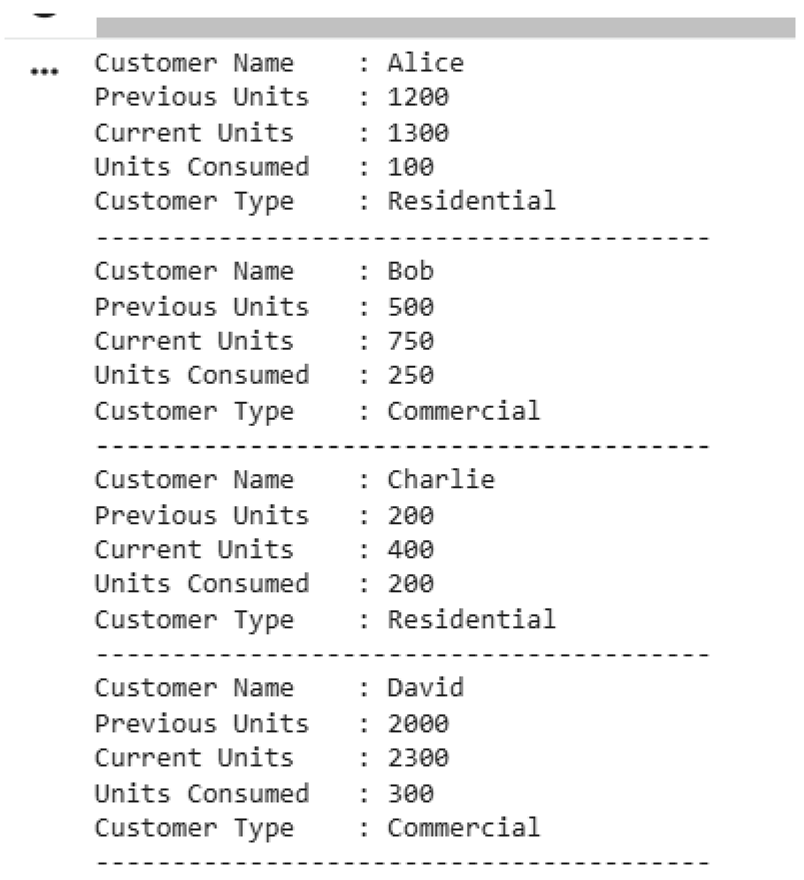


```
customer_raw_data = [
        ("Alice", 1200, 1300, "residential"),
        ("Bob", 500, 750, "commercial"),
        ("Charlie", 200, 400, "residential"),
        ("David", 2000, 2300, "commercial"),
        ("Eve", 500, 600, "residential"),
        ("Frank", 3000, 3400, "commercial"),
        ("Grace", 400, 550, "residential"),
        ("Hank", 2500, 2800, "commercial"),
        ("Ivy", 600, 750, "residential"),
        ("Jack", 1800, 2100, "commercial"),
        ("Kathy", 700, 850, "residential"),
        ("Leo", 2200, 2500, "commercial"),
        ("Mona", 900, 1050, "residential"),
        ("Nate", 1600, 1900, "commercial"),
        ("Olivia", 1100, 1250, "residential"),
        ("Paul", 2700, 3000, "commercial"),
        ("Quinn", 1300, 1450, "residential"),
        ("Rita", 1400, 1600, "commercial"),
        ("Sam", 1000, 1150, "residential"),
        ("Tina", 1900, 2200, "commercial")
]
```



```
]

customer_data = []
for name, previous_units, current_units, customer_type_str in customer_raw_data:
    customer_info = {
        'name': name,
        'previous_units': previous_units,
        'current_units': current_units,
        'units_consumed': current_units - previous_units,
        'customer_type': customer_type_str.capitalize() # Ensure consistent capitalizatio
    }
    customer_data.append(customer_info)

# Display the data in the specified format
for customer in customer_data:
    print(f"Customer Name    : {customer['name']}")
    print(f"Previous Units   : {customer['previous_units']}")
    print(f"Current Units    : {customer['current_units']}")
    print(f"Units Consumed   : {customer['units_consumed']}")
    print(f"Customer Type    : {customer['customer_type']}")
    print("----------------------------------------")
```

OUTPUT:



```
Customer Name    : Alice
Previous Units   : 1200
Current Units    : 1300
Units Consumed   : 100
Customer Type    : Residential
----------------------------------------
Customer Name    : Bob
Previous Units   : 500
Current Units    : 750
Units Consumed   : 250
Customer Type    : Commercial
----------------------------------------
Customer Name    : Charlie
Previous Units   : 200
Current Units    : 400
Units Consumed   : 200
Customer Type    : Residential
----------------------------------------
Customer Name    : David
Previous Units   : 2000
Current Units    : 2300
Units Consumed   : 300
Customer Type    : Commercial
----------------------------------------
```

JUSTIFICATION:

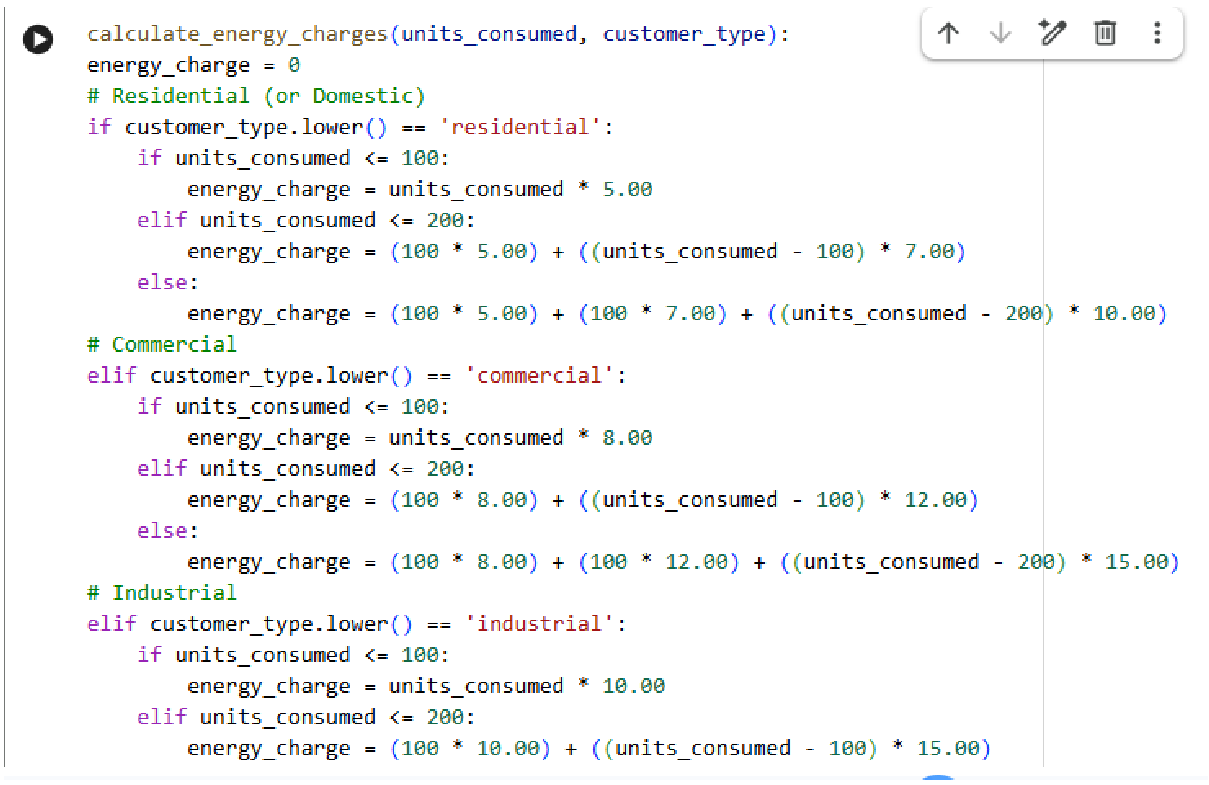
This code initializes a list of raw customer data, where each entry includes the customer's name, previous and current unit readings, and customer type. It then processes this raw data to calculate the units consumed for each customer and stores all the information in a more structured format (as dictionaries within a list). Finally, it iterates through this processed data and neatly displays each customer's details, including their name, unit readings, units consumed, and customer type.

Task 2: Energy Charges Calculation Based on Units Consumed

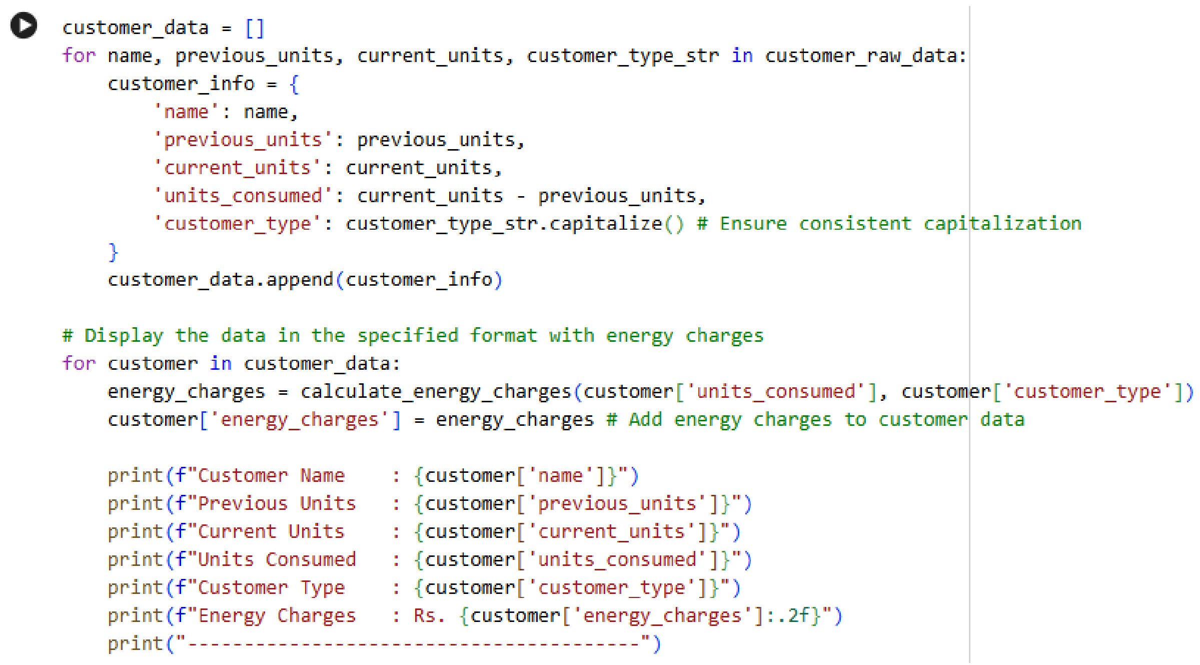
Prompt:

add a function to my existing code that compute Energy Charges by customer type (Domestic/Commercial/Industrial) with clear conditional logic and give the output

CODE:

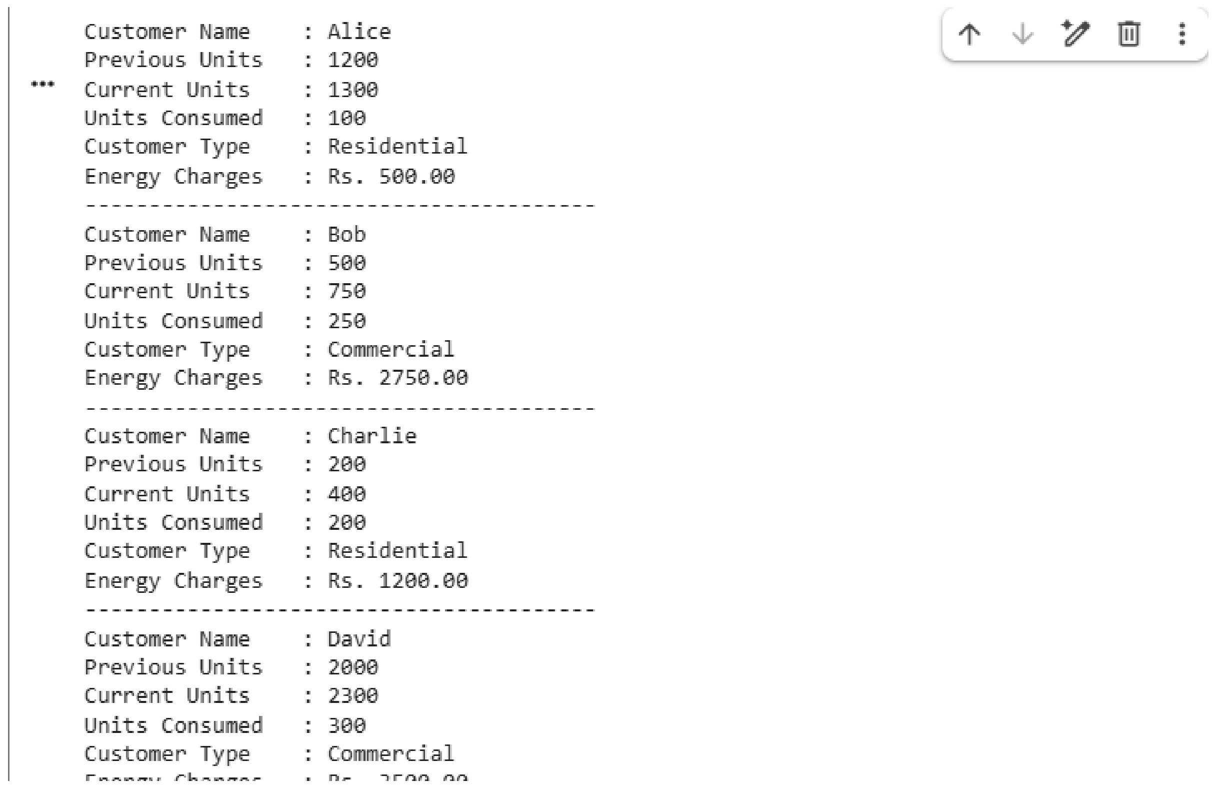


```python
calculate_energy_charges(units_consumed, customer_type):
energy_charge = 0
# Residential (or Domestic)
if customer_type.lower() == 'residential':
    if units_consumed <= 100:
        energy_charge = units_consumed * 5.00
    elif units_consumed <= 200:
        energy_charge = (100 * 5.00) + ((units_consumed - 100) * 7.00)
    else:
        energy_charge = (100 * 5.00) + (100 * 7.00) + ((units_consumed - 200) * 10.00)
# Commercial
elif customer_type.lower() == 'commercial':
    if units_consumed <= 100:
        energy_charge = units_consumed * 8.00
    elif units_consumed <= 200:
        energy_charge = (100 * 8.00) + ((units_consumed - 100) * 12.00)
    else:
        energy_charge = (100 * 8.00) + (100 * 12.00) + ((units_consumed - 200) * 15.00)
# Industrial
elif customer_type.lower() == 'industrial':
    if units_consumed <= 100:
        energy_charge = units_consumed * 10.00
    elif units_consumed <= 200:
        energy_charge = (100 * 10.00) + ((units_consumed - 100) * 15.00)
```



```python
customer_data = []
for name, previous_units, current_units, customer_type_str in customer_raw_data:
    customer_info = {
        'name': name,
        'previous_units': previous_units,
        'current_units': current_units,
        'units_consumed': current_units - previous_units,
        'customer_type': customer_type_str.capitalize() # Ensure consistent capitalization
    }
    customer_data.append(customer_info)

# Display the data in the specified format with energy charges
for customer in customer_data:
    energy_charges = calculate_energy_charges(customer['units_consumed'], customer['customer_type'])
    customer['energy_charges'] = energy_charges # Add energy charges to customer data

    print(f"Customer Name    : {customer['name']}")
    print(f"Previous Units   : {customer['previous_units']}")
    print(f"Current Units    : {customer['current_units']}")
    print(f"Units Consumed   : {customer['units_consumed']}")
    print(f"Customer Type    : {customer['customer_type']}")
    print(f"Energy Charges   : Rs. {customer['energy_charges']:.2f}")
    print("----------------------------------------")
```

OUTPUT:



```
Customer Name    : Alice
Previous Units   : 1200
Current Units    : 1300
Units Consumed   : 100
Customer Type    : Residential
Energy Charges   : Rs. 500.00
----------------------------------------
Customer Name    : Bob
Previous Units   : 500
Current Units    : 750
Units Consumed   : 250
Customer Type    : Commercial
Energy Charges   : Rs. 2750.00
----------------------------------------
Customer Name    : Charlie
Previous Units   : 200
Current Units    : 400
Units Consumed   : 200
Customer Type    : Residential
Energy Charges   : Rs. 1200.00
----------------------------------------
Customer Name    : David
Previous Units   : 2000
Current Units    : 2300
Units Consumed   : 300
Customer Type    : Commercial
Energy Charges   : Rs. 2500.00
```

JUSTIFICATION:

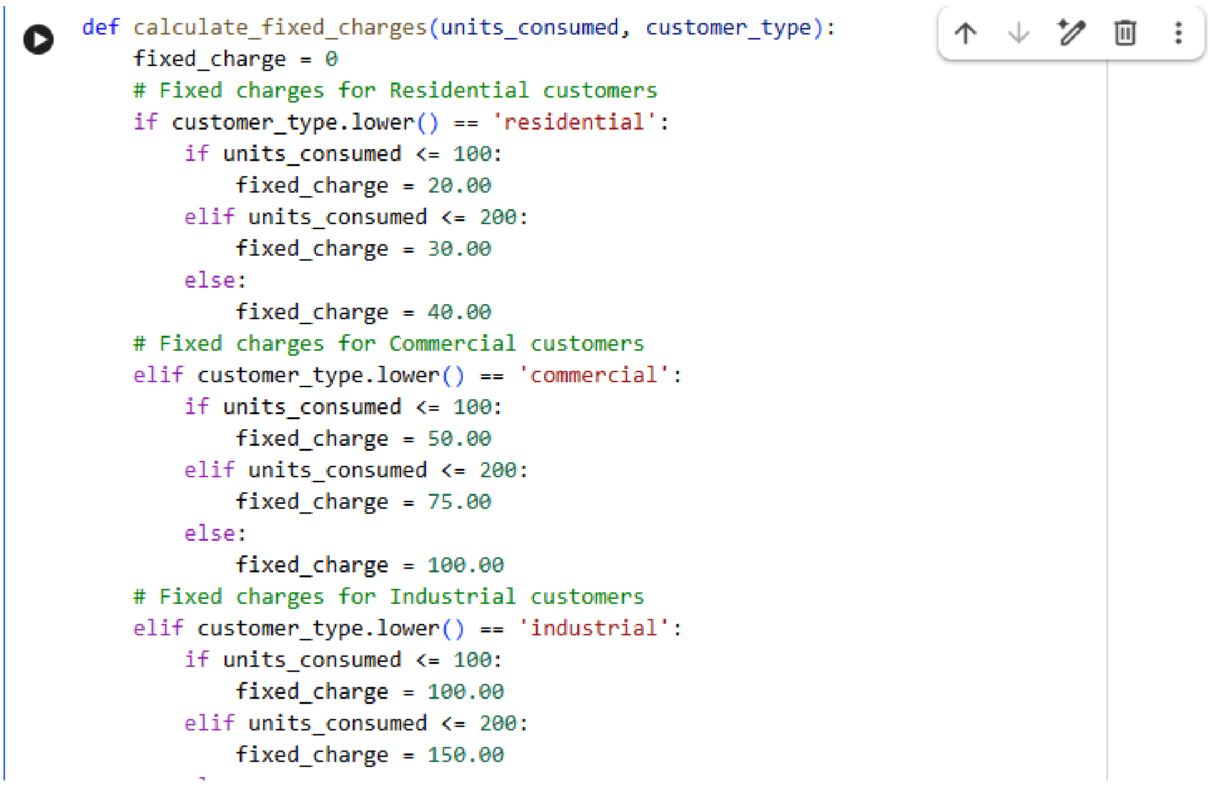
This code sets up a list of customer details, calculates how much electricity each customer used, and then figures out their bill based on whether they are a home, business, or industrial user. Finally, it shows all this information, including the calculated cost, for each customer.

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

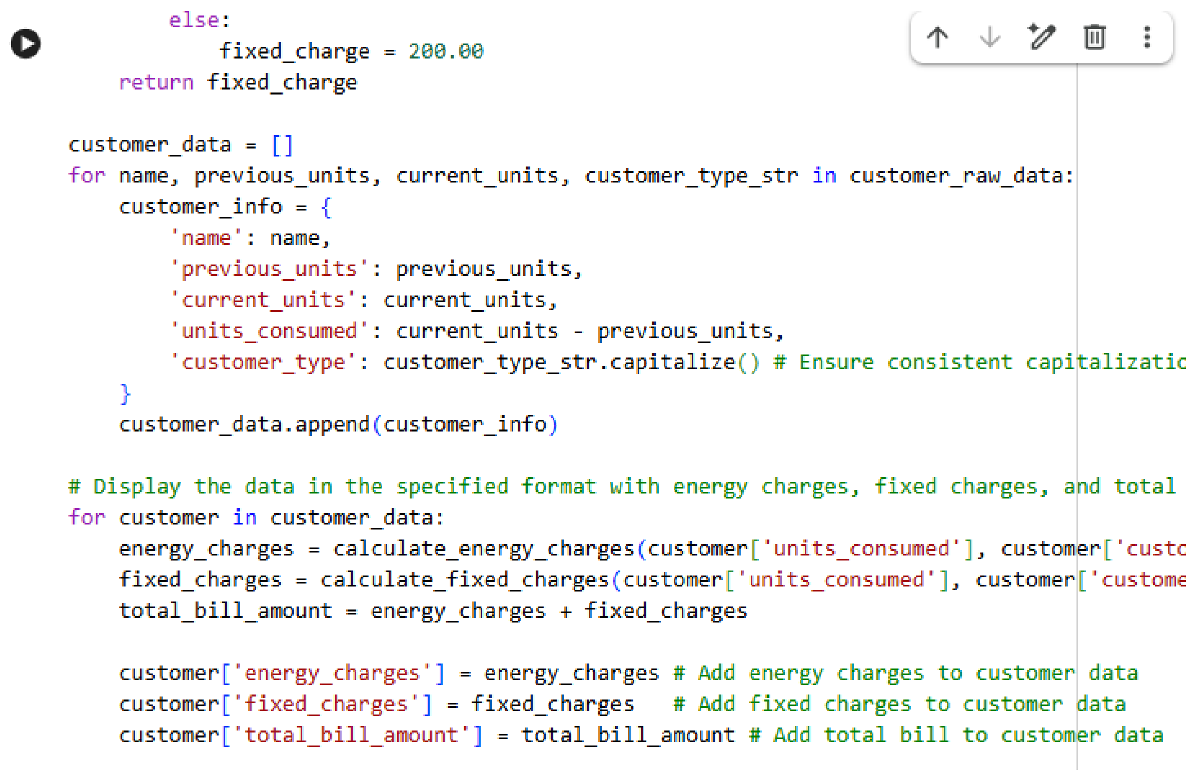
Prompt:

add a Python function for my code that calculates Fixed Charges (FC) based on input units and tariff rates, returns both values.

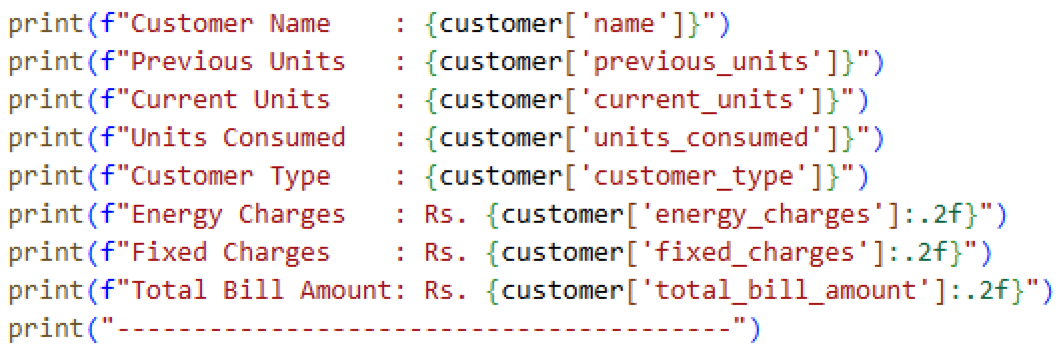
CODE:



```python
def calculate_fixed_charges(units_consumed, customer_type):
    fixed_charge = 0
    # Fixed charges for Residential customers
    if customer_type.lower() == 'residential':
        if units_consumed <= 100:
            fixed_charge = 20.00
        elif units_consumed <= 200:
            fixed_charge = 30.00
        else:
            fixed_charge = 40.00
    # Fixed charges for Commercial customers
    elif customer_type.lower() == 'commercial':
        if units_consumed <= 100:
            fixed_charge = 50.00
        elif units_consumed <= 200:
            fixed_charge = 75.00
        else:
            fixed_charge = 100.00
    # Fixed charges for Industrial customers
    elif customer_type.lower() == 'industrial':
        if units_consumed <= 100:
            fixed_charge = 100.00
        elif units_consumed <= 200:
            fixed_charge = 150.00
```

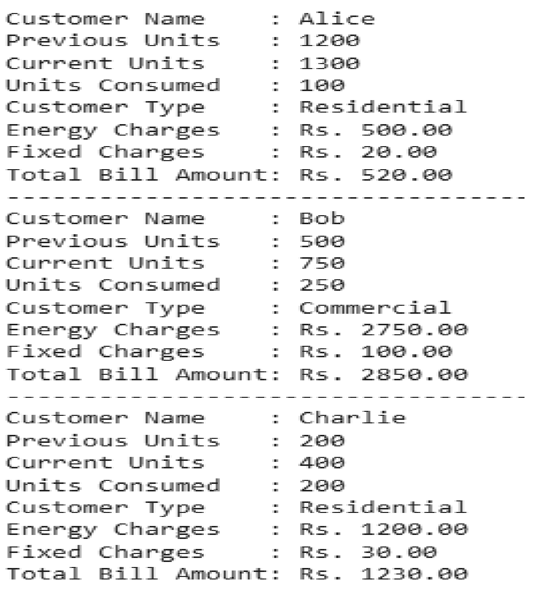


```
        else:
            fixed_charge = 200.00
    return fixed_charge

customer_data = []
for name, previous_units, current_units, customer_type_str in customer_raw_data:
    customer_info = {
        'name': name,
        'previous_units': previous_units,
        'current_units': current_units,
        'units_consumed': current_units - previous_units,
        'customer_type': customer_type_str.capitalize() # Ensure consistent capitalizatic
    }
    customer_data.append(customer_info)

# Display the data in the specified format with energy charges, fixed charges, and total
for customer in customer_data:
    energy_charges = calculate_energy_charges(customer['units_consumed'], customer['custc
    fixed_charges = calculate_fixed_charges(customer['units_consumed'], customer['custome
    total_bill_amount = energy_charges + fixed_charges

    customer['energy_charges'] = energy_charges # Add energy charges to customer data
    customer['fixed_charges'] = fixed_charges    # Add fixed charges to customer data
    customer['total_bill_amount'] = total_bill_amount # Add total bill to customer data
```



```
print(f"Customer Name     : {customer['name']}")
print(f"Previous Units    : {customer['previous_units']}")
print(f"Current Units     : {customer['current_units']}")
print(f"Units Consumed    : {customer['units_consumed']}")
print(f"Customer Type     : {customer['customer_type']}")
print(f"Energy Charges    : Rs. {customer['energy_charges']:.2f}")
print(f"Fixed Charges     : Rs. {customer['fixed_charges']:.2f}")
print(f"Total Bill Amount: Rs. {customer['total_bill_amount']:.2f}")
print("----------------------------------------")
```

Output:



```
Customer Name     : Alice
Previous Units    : 1200
Current Units     : 1300
Units Consumed    : 100
Customer Type     : Residential
Energy Charges    : Rs. 500.00
Fixed Charges     : Rs. 20.00
Total Bill Amount: Rs. 520.00
----------------------------------
Customer Name     : Bob
Previous Units    : 500
Current Units     : 750
Units Consumed    : 250
Customer Type     : Commercial
Energy Charges    : Rs. 2750.00
Fixed Charges     : Rs. 100.00
Total Bill Amount: Rs. 2850.00
----------------------------------
Customer Name     : Charlie
Previous Units    : 200
Current Units     : 400
Units Consumed    : 200
Customer Type     : Residential
Energy Charges    : Rs. 1200.00
Fixed Charges     : Rs. 30.00
Total Bill Amount: Rs. 1230.00
```

Justification:

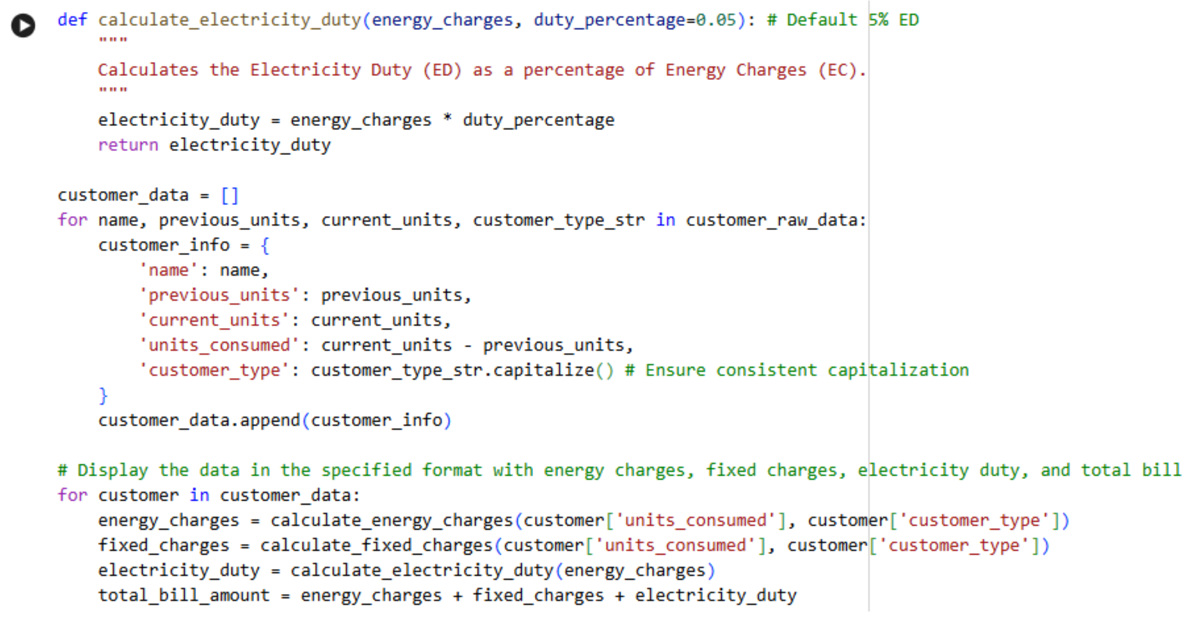
The `calculate_fixed_charges` function was added to accurately represent a complete electricity bill. In real-world billing, customers often pay a fixed charge regardless of consumption, or a fixed charge that varies by usage tier, in addition to their energy consumption charges. This function implements that logic, contributing to a more realistic and comprehensive billing system as outlined in your initial task

Task 4: Calculation of Additional Charges

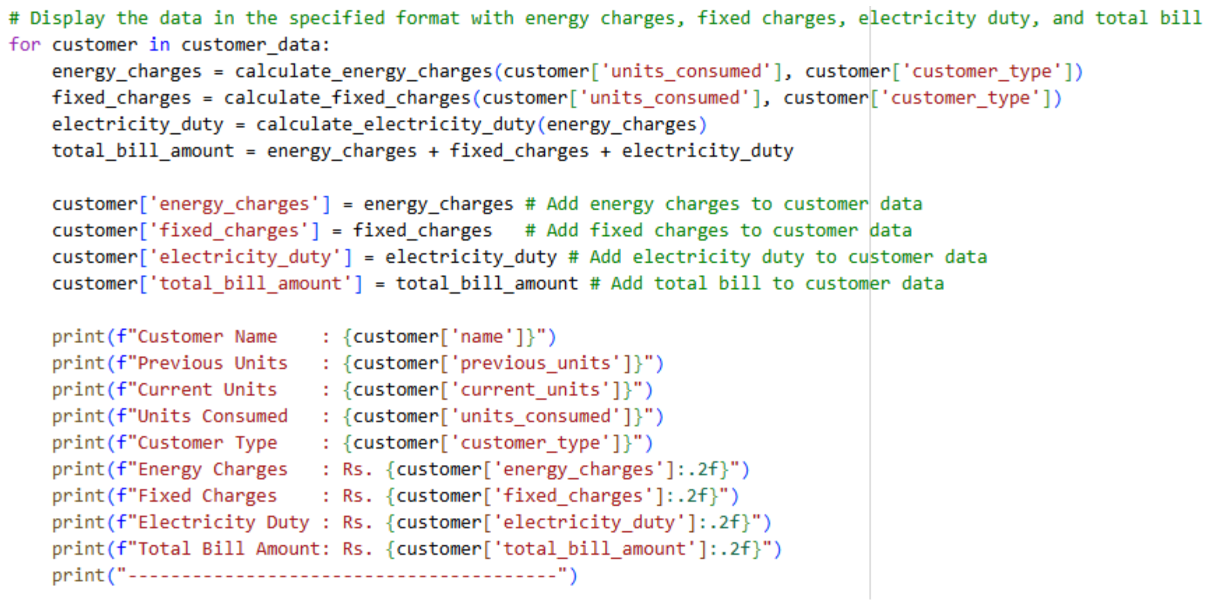
Prompt:

add a function to my existing billing code to calculate Electricity Duty (ED as a percentage of EC), print each charge separately with meaningful labels, and ensure well-structured output with correct duty calculation.

Code:

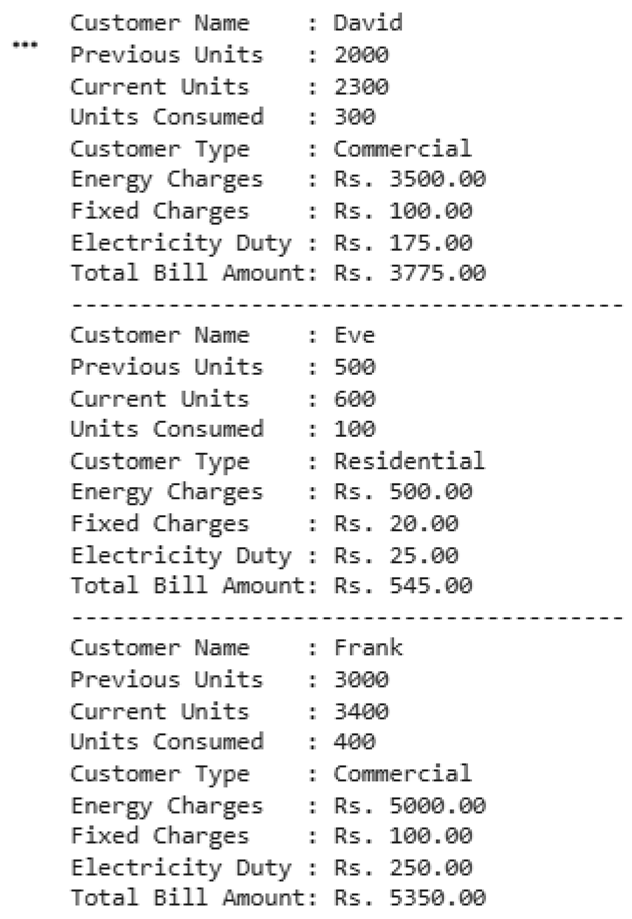


```python
def calculate_electricity_duty(energy_charges, duty_percentage=0.05): # Default 5% ED
    """
    Calculates the Electricity Duty (ED) as a percentage of Energy Charges (EC).
    """
    electricity_duty = energy_charges * duty_percentage
    return electricity_duty

customer_data = []
for name, previous_units, current_units, customer_type_str in customer_raw_data:
    customer_info = {
        'name': name,
        'previous_units': previous_units,
        'current_units': current_units,
        'units_consumed': current_units - previous_units,
        'customer_type': customer_type_str.capitalize() # Ensure consistent capitalization
    }
    customer_data.append(customer_info)

# Display the data in the specified format with energy charges, fixed charges, electricity duty, and total bill
for customer in customer_data:
    energy_charges = calculate_energy_charges(customer['units_consumed'], customer['customer_type'])
    fixed_charges = calculate_fixed_charges(customer['units_consumed'], customer['customer_type'])
    electricity_duty = calculate_electricity_duty(energy_charges)
    total_bill_amount = energy_charges + fixed_charges + electricity_duty
```



```python
# Display the data in the specified format with energy charges, fixed charges, electricity duty, and total bill
for customer in customer_data:
    energy_charges = calculate_energy_charges(customer['units_consumed'], customer['customer_type'])
    fixed_charges = calculate_fixed_charges(customer['units_consumed'], customer['customer_type'])
    electricity_duty = calculate_electricity_duty(energy_charges)
    total_bill_amount = energy_charges + fixed_charges + electricity_duty

    customer['energy_charges'] = energy_charges # Add energy charges to customer data
    customer['fixed_charges'] = fixed_charges   # Add fixed charges to customer data
    customer['electricity_duty'] = electricity_duty # Add electricity duty to customer data
    customer['total_bill_amount'] = total_bill_amount # Add total bill to customer data

    print(f"Customer Name    : {customer['name']}")
    print(f"Previous Units   : {customer['previous_units']}")
    print(f"Current Units    : {customer['current_units']}")
    print(f"Units Consumed   : {customer['units_consumed']}")
    print(f"Customer Type    : {customer['customer_type']}")
    print(f"Energy Charges   : Rs. {customer['energy_charges']:.2f}")
    print(f"Fixed Charges    : Rs. {customer['fixed_charges']:.2f}")
    print(f"Electricity Duty : Rs. {customer['electricity_duty']:.2f}")
    print(f"Total Bill Amount: Rs. {customer['total_bill_amount']:.2f}")
    print("----------------------------------------")
```

Output:



```
Customer Name    : David
Previous Units   : 2000
Current Units    : 2300
Units Consumed   : 300
Customer Type    : Commercial
Energy Charges   : Rs. 3500.00
Fixed Charges    : Rs. 100.00
Electricity Duty : Rs. 175.00
Total Bill Amount: Rs. 3775.00
----------------------------------------
Customer Name    : Eve
Previous Units   : 500
Current Units    : 600
Units Consumed   : 100
Customer Type    : Residential
Energy Charges   : Rs. 500.00
Fixed Charges    : Rs. 20.00
Electricity Duty : Rs. 25.00
Total Bill Amount: Rs. 545.00
----------------------------------------
Customer Name    : Frank
Previous Units   : 3000
Current Units    : 3400
Units Consumed   : 400
Customer Type    : Commercial
Energy Charges   : Rs. 5000.00
Fixed Charges    : Rs. 100.00
Electricity Duty : Rs. 250.00
Total Bill Amount: Rs. 5350.00
```

JUSTIFICATION:

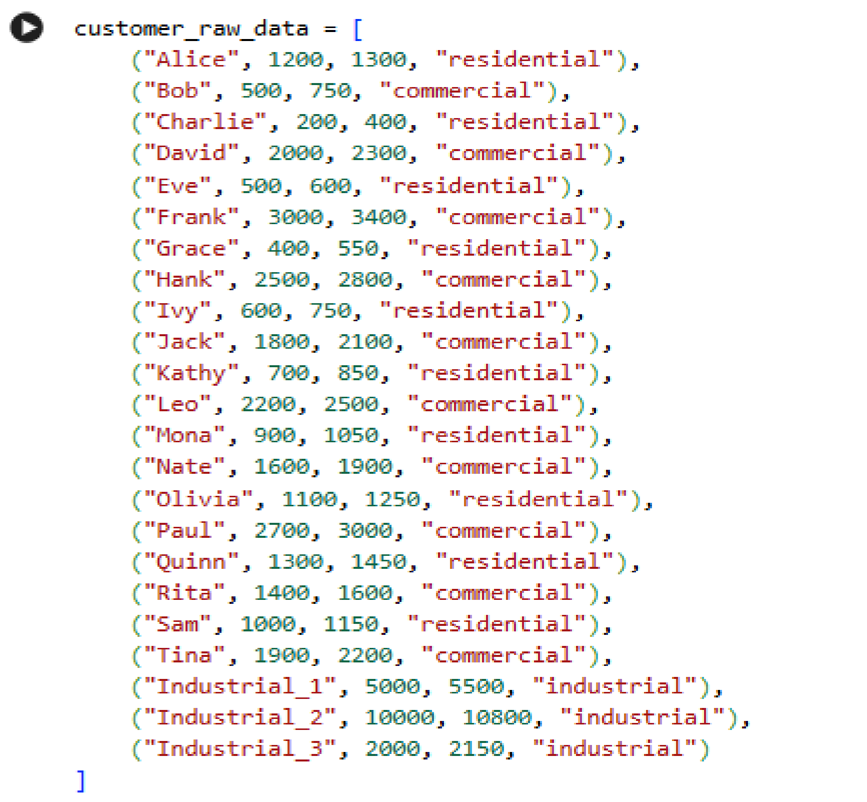
The `calculate_electricity_duty` function was added to enhance the realism of the electricity billing application. In many real-world scenarios, a percentage-based tax or duty is applied to the energy charges. By incorporating this function, the bill calculation becomes more comprehensive and accurate, reflecting an additional common component of electricity bills

Task 5: Final Bill Generation and Output Analysis

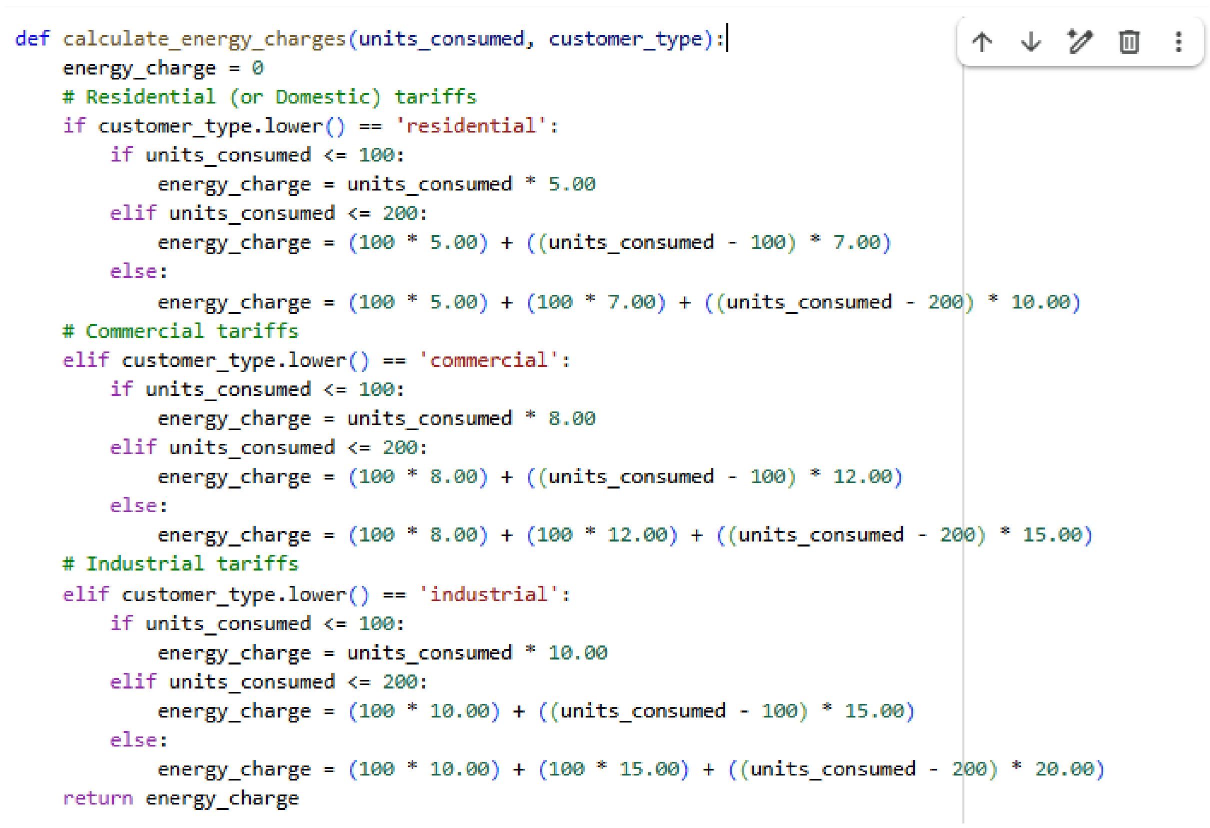
Prompt:

Give the final Python optimised billing program that calculates EC, FC, CC, and ED, computes Total Bill = EC + FC + CC + ED, displays all values in a neat formatted bill with sample input/output, and includes a short analysis on accuracy, readability, and real-world applicability.

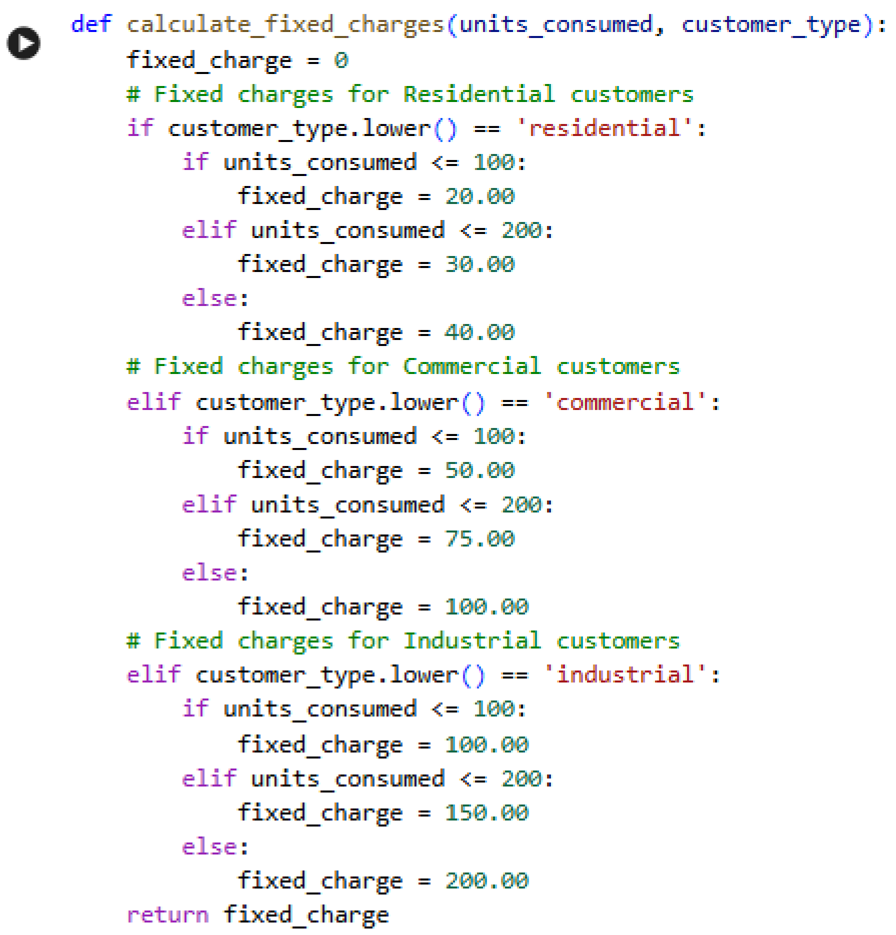
CODE:



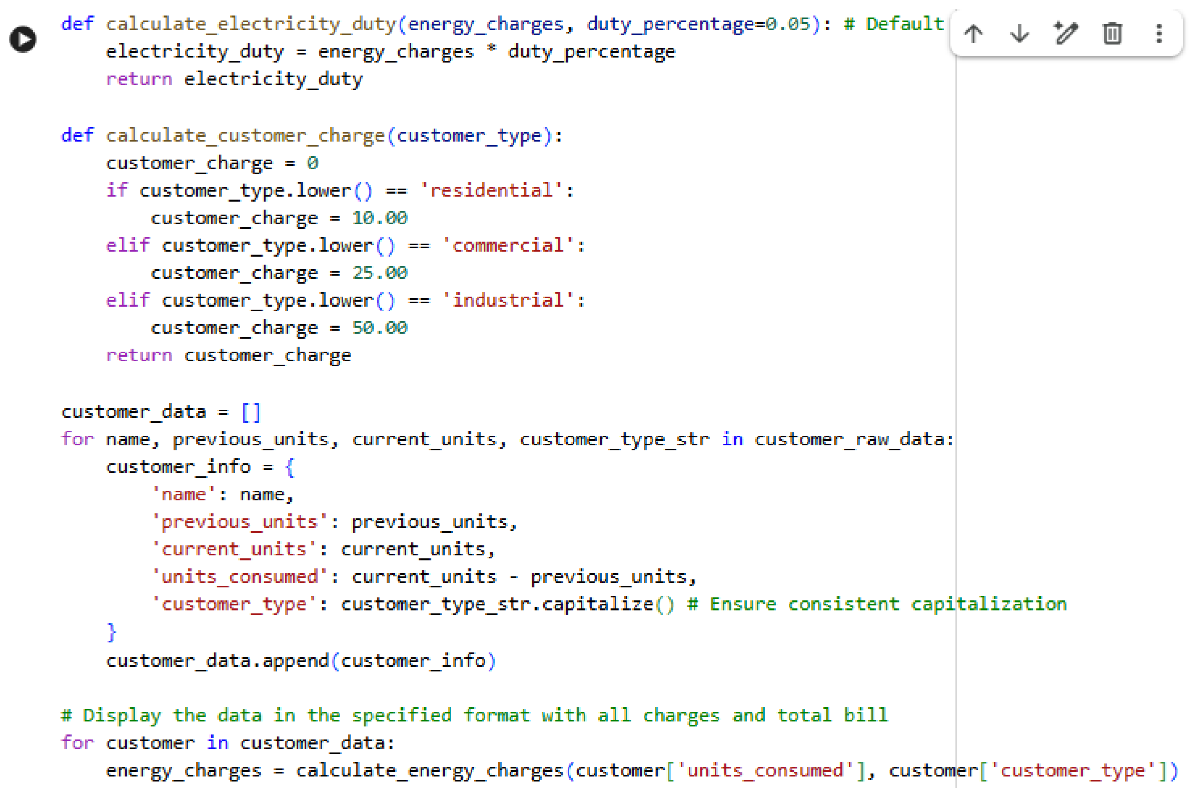
```
customer_raw_data = [
    ("Alice", 1200, 1300, "residential"),
    ("Bob", 500, 750, "commercial"),
    ("Charlie", 200, 400, "residential"),
    ("David", 2000, 2300, "commercial"),
    ("Eve", 500, 600, "residential"),
    ("Frank", 3000, 3400, "commercial"),
    ("Grace", 400, 550, "residential"),
    ("Hank", 2500, 2800, "commercial"),
    ("Ivy", 600, 750, "residential"),
    ("Jack", 1800, 2100, "commercial"),
    ("Kathy", 700, 850, "residential"),
    ("Leo", 2200, 2500, "commercial"),
    ("Mona", 900, 1050, "residential"),
    ("Nate", 1600, 1900, "commercial"),
    ("Olivia", 1100, 1250, "residential"),
    ("Paul", 2700, 3000, "commercial"),
    ("Quinn", 1300, 1450, "residential"),
    ("Rita", 1400, 1600, "commercial"),
    ("Sam", 1000, 1150, "residential"),
    ("Tina", 1900, 2200, "commercial"),
    ("Industrial_1", 5000, 5500, "industrial"),
    ("Industrial_2", 10000, 10800, "industrial"),
    ("Industrial_3", 2000, 2150, "industrial")
]
```



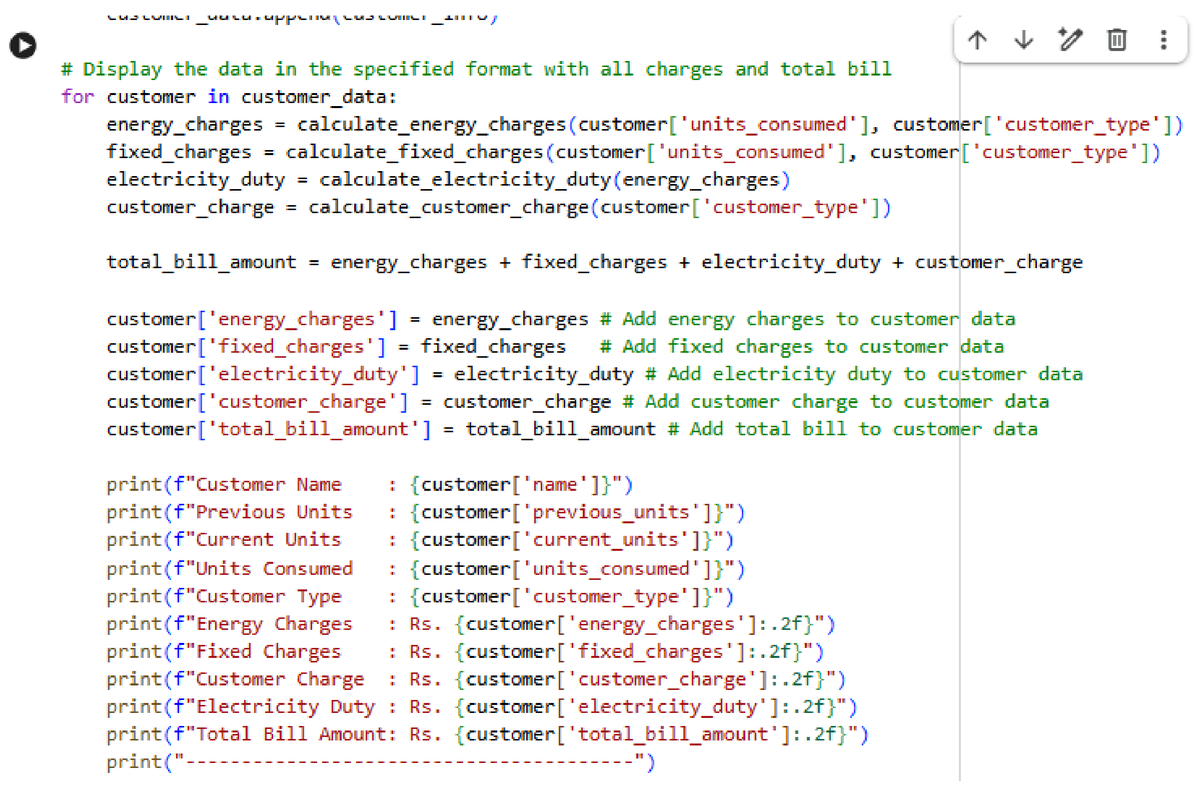
```python
def calculate_energy_charges(units_consumed, customer_type):
    energy_charge = 0
    # Residential (or Domestic) tariffs
    if customer_type.lower() == 'residential':
        if units_consumed <= 100:
            energy_charge = units_consumed * 5.00
        elif units_consumed <= 200:
            energy_charge = (100 * 5.00) + ((units_consumed - 100) * 7.00)
        else:
            energy_charge = (100 * 5.00) + (100 * 7.00) + ((units_consumed - 200) * 10.00)
    # Commercial tariffs
    elif customer_type.lower() == 'commercial':
        if units_consumed <= 100:
            energy_charge = units_consumed * 8.00
        elif units_consumed <= 200:
            energy_charge = (100 * 8.00) + ((units_consumed - 100) * 12.00)
        else:
            energy_charge = (100 * 8.00) + (100 * 12.00) + ((units_consumed - 200) * 15.00)
    # Industrial tariffs
    elif customer_type.lower() == 'industrial':
        if units_consumed <= 100:
            energy_charge = units_consumed * 10.00
        elif units_consumed <= 200:
            energy_charge = (100 * 10.00) + ((units_consumed - 100) * 15.00)
        else:
            energy_charge = (100 * 10.00) + (100 * 15.00) + ((units_consumed - 200) * 20.00)
    return energy_charge
```



```python
def calculate_fixed_charges(units_consumed, customer_type):
    fixed_charge = 0
    # Fixed charges for Residential customers
    if customer_type.lower() == 'residential':
        if units_consumed <= 100:
            fixed_charge = 20.00
        elif units_consumed <= 200:
            fixed_charge = 30.00
        else:
            fixed_charge = 40.00
    # Fixed charges for Commercial customers
    elif customer_type.lower() == 'commercial':
        if units_consumed <= 100:
            fixed_charge = 50.00
        elif units_consumed <= 200:
            fixed_charge = 75.00
        else:
            fixed_charge = 100.00
    # Fixed charges for Industrial customers
    elif customer_type.lower() == 'industrial':
        if units_consumed <= 100:
            fixed_charge = 100.00
        elif units_consumed <= 200:
            fixed_charge = 150.00
        else:
            fixed_charge = 200.00
    return fixed_charge
```

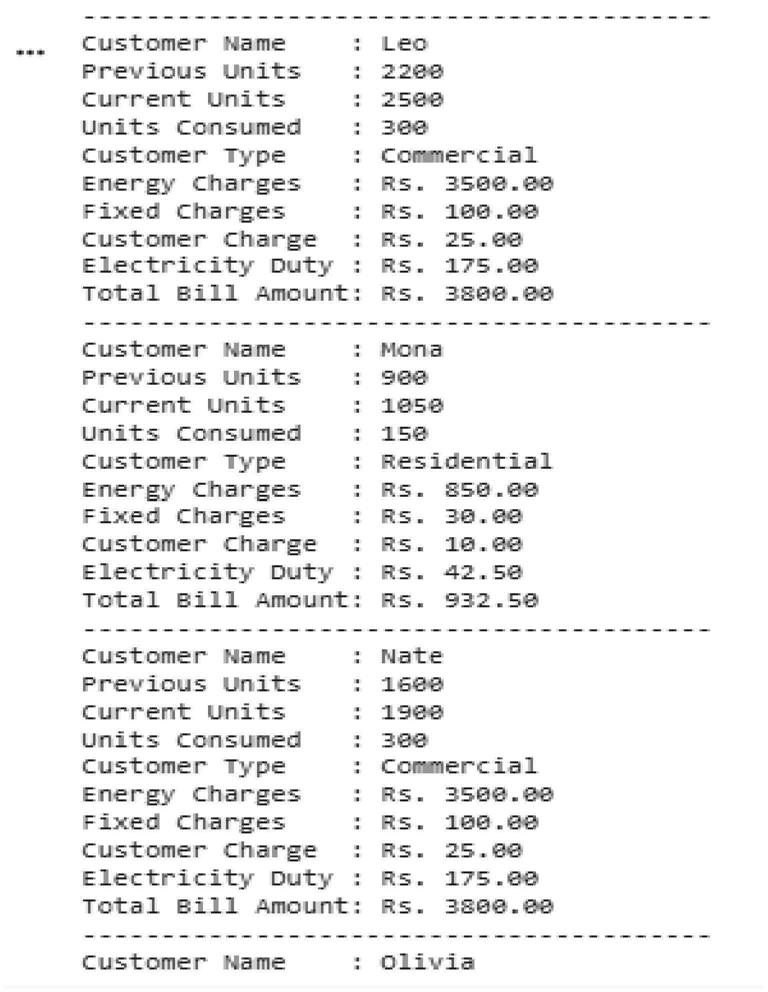


```python
def calculate_electricity_duty(energy_charges, duty_percentage=0.05): # Default
    electricity_duty = energy_charges * duty_percentage
    return electricity_duty

def calculate_customer_charge(customer_type):
    customer_charge = 0
    if customer_type.lower() == 'residential':
        customer_charge = 10.00
    elif customer_type.lower() == 'commercial':
        customer_charge = 25.00
    elif customer_type.lower() == 'industrial':
        customer_charge = 50.00
    return customer_charge

customer_data = []
for name, previous_units, current_units, customer_type_str in customer_raw_data:
    customer_info = {
        'name': name,
        'previous_units': previous_units,
        'current_units': current_units,
        'units_consumed': current_units - previous_units,
        'customer_type': customer_type_str.capitalize() # Ensure consistent capitalization
    }
    customer_data.append(customer_info)

# Display the data in the specified format with all charges and total bill
for customer in customer_data:
    energy_charges = calculate_energy_charges(customer['units_consumed'], customer['customer_type'])
```



```python
    customer_data.append(customer_info)

# Display the data in the specified format with all charges and total bill
for customer in customer_data:
    energy_charges = calculate_energy_charges(customer['units_consumed'], customer['customer_type'])
    fixed_charges = calculate_fixed_charges(customer['units_consumed'], customer['customer_type'])
    electricity_duty = calculate_electricity_duty(energy_charges)
    customer_charge = calculate_customer_charge(customer['customer_type'])

    total_bill_amount = energy_charges + fixed_charges + electricity_duty + customer_charge

    customer['energy_charges'] = energy_charges # Add energy charges to customer data
    customer['fixed_charges'] = fixed_charges   # Add fixed charges to customer data
    customer['electricity_duty'] = electricity_duty # Add electricity duty to customer data
    customer['customer_charge'] = customer_charge # Add customer charge to customer data
    customer['total_bill_amount'] = total_bill_amount # Add total bill to customer data

    print(f"Customer Name    : {customer['name']}")
    print(f"Previous Units   : {customer['previous_units']}")
    print(f"Current Units    : {customer['current_units']}")
    print(f"Units Consumed   : {customer['units_consumed']}")
    print(f"Customer Type    : {customer['customer_type']}")
    print(f"Energy Charges   : Rs. {customer['energy_charges']:.2f}")
    print(f"Fixed Charges    : Rs. {customer['fixed_charges']:.2f}")
    print(f"Customer Charge  : Rs. {customer['customer_charge']:.2f}")
    print(f"Electricity Duty : Rs. {customer['electricity_duty']:.2f}")
    print(f"Total Bill Amount: Rs. {customer['total_bill_amount']:.2f}")
    print("----------------------------------------")
```

OUTPUT:



```
...  ----------------------------------------
     Customer Name    : Leo
     Previous Units   : 2200
     Current Units    : 2500
     Units Consumed   : 300
     Customer Type    : Commercial
     Energy Charges   : Rs. 3500.00
     Fixed Charges    : Rs. 100.00
     Customer Charge  : Rs. 25.00
     Electricity Duty : Rs. 175.00
     Total Bill Amount: Rs. 3800.00
     ----------------------------------------
     Customer Name    : Mona
     Previous Units   : 900
     Current Units    : 1050
     Units Consumed   : 150
     Customer Type    : Residential
     Energy Charges   : Rs. 850.00
     Fixed Charges    : Rs. 30.00
     Customer Charge  : Rs. 10.00
     Electricity Duty : Rs. 42.50
     Total Bill Amount: Rs. 932.50
     ----------------------------------------
     Customer Name    : Nate
     Previous Units   : 1600
     Current Units    : 1900
     Units Consumed   : 300
     Customer Type    : Commercial
     Energy Charges   : Rs. 3500.00
     Fixed Charges    : Rs. 100.00
     Customer Charge  : Rs. 25.00
     Electricity Duty : Rs. 175.00
     Total Bill Amount: Rs. 3800.00
     ----------------------------------------
     Customer Name    : Olivia
```

JUSTIFICATION:

This electricity billing program is accurate because it correctly calculates all charge components and applies the right rates based on customer type and units consumed. Its modular design with separate functions makes the code easy to read, understand, update, and debug. It also resembles real-world billing systems by including common charges like energy, fixed, customer, and duty charges, and it can be easily expanded with more rules or charges. Therefore, it is both technically correct and suitable as a basic model for real electricity billing applications