SYED NAZIYA PARVEEN

2303A52394

BATCH-42

TASK 1: Generate a simple Python program to read previous units, current units, and customer type, then calculate units consumed without using functions.

```
customers = [
    ("C1", "Domestic", 120, 180),
    ("C2", "Domestic", 340, 420),
    ("C3", "Domestic", 500, 650),
    ("C4", "Domestic", 50, 160),
    ("C5", "Domestic", 200, 260),
    ("C6", "Commercial", 1000, 1120),
    ("C7", "Commercial", 450, 600),
    ("C8", "Commercial", 800, 880),
    ("C9", "Commercial", 300, 420),
    ("C10", "Commercial", 1500, 1650),
    ("C11", "Domestic", 600, 690),
    ("C12", "Domestic", 50, 170),
    ("C13", "Commercial", 200, 350),
    ("C14", "Domestic", 900, 980),
    ("C15", "Commercial", 700, 820)
]

print("Electricity Billing Details")
print("-----------------------------------------------------------------")
print("Cust  Type        PU    CU    Units   Bill Amount (₹)")
print("-----------------------------------------------------------------")

for customer in customers:
    cust_id, cust_type, PU, CU = customer
    units_consumed = CU - PU
    bill_amount = 0

    if cust_type == "Domestic":
        if units_consumed <= 100:
            bill_amount = units_consumed * 3
        else:
            bill_amount = (100 * 3) + ((units_consumed - 100) * 5)

    elif cust_type == "Commercial":
        if units_consumed <= 100:
            bill_amount = units_consumed * 6
        else:
            bill_amount = (100 * 6) + ((units_consumed - 100) * 8)

    print(cust_id, " ", cust_type.ljust(10), PU, " ", CU, "   ",
          units_consumed, "      ₹", bill_amount)

print("-----------------------------------------------------------------")
```

OUTPUT:

```
Electricity Billing Details
-----------------------------------------------------------------
Cust  Type         PU    CU    Units   Bill Amount (₹)
-----------------------------------------------------------------
C1   Domestic   120   180     60       ₹ 180
C2   Domestic   340   420     80       ₹ 240
C3   Domestic   500   650     150       ₹ 550
C4   Domestic   50   160     110       ₹ 350
C5   Domestic   200   260     60       ₹ 180
C6   Commercial 1000   1120     120       ₹ 760
C7   Commercial 450   600     150       ₹ 1000
C8   Commercial 800   880     80       ₹ 480
C9   Commercial 300   420     120       ₹ 760
C10   Commercial 1500   1650     150       ₹ 1000
C11   Domestic   600   690     90       ₹ 270
C12   Domestic   50   170     120       ₹ 400
C13   Commercial 200   350     150       ₹ 1000
C14   Domestic   900   980     80       ₹ 240
C15   Commercial 700   820     120       ₹ 760
```

JUSTIFICATION:

This code reads predefined customer data and calculates units consumed.
It applies slab-based tariff logic using conditional statements without functions, fulfilling basic billing requirements.

TASK 2:

Extend the program to calculate energy charges based on units consumed

using conditional statements for domestic, commercial, and industrial users.

```
customers = [
    ("C1", "Domestic", 120, 180),
    ("C2", "Domestic", 340, 420),
    ("C3", "Domestic", 500, 650),
    ("C4", "Domestic", 50, 160),
    ("C5", "Domestic", 200, 260),
    ("C6", "Commercial", 1000, 1120),
    ("C7", "Commercial", 450, 600),
    ("C8", "Commercial", 800, 880),
    ("C9", "Commercial", 300, 420),
    ("C10", "Commercial", 1500, 1650),
    ("C11", "Domestic", 600, 690),
    ("C12", "Domestic", 50, 170),
    ("C13", "Commercial", 200, 350),
    ("C14", "Domestic", 900, 980),
    ("C15", "Commercial", 700, 820)
]

rates = {
    "Domestic": (3, 5),
    "Commercial": (6, 8),
    "Industrial": (8, 10)
}
```

```
print("Energy Billing Details")
print("-----------------------------------------------------------------")
print("Cust  Type        PU    CU    Units   Energy Charges (₹)")
print("-----------------------------------------------------------------")

for cust_id, cust_type, PU, CU in customers:
    units = CU - PU
    r1, r2 = rates[cust_type]

    # Formula-based calculation
    ec = (min(units, 100) * r1) + (max(units - 100, 0) * r2)

    print(f"{cust_id:<4} {cust_type:<11} {PU:<5} {CU:<5} {units:<6} ₹{ec}")

print("-----------------------------------------------------------------")
```

OUTPUT

```
Energy Billing Details
-----------------------------------------------------------------
Cust  Type        PU    CU    Units   Energy Charges (₹)
-----------------------------------------------------------------
C1   Domestic    120   180   60     ₹180
C2   Domestic    340   420   80     ₹240
C3   Domestic    500   650   150    ₹550
C4   Domestic    50    160   110    ₹350
C5   Domestic    200   260   60     ₹180
C6   Commercial  1000  1120  120    ₹760
C7   Commercial  450   600   150    ₹1000
C8   Commercial  800   880   80     ₹480
C9   Commercial  300   420   120    ₹760
C10  Commercial  1500  1650  150    ₹1000
C11  Domestic    600   690   90     ₹270
C12  Domestic    50    170   120    ₹400
C13  Commercial  200   350   150    ₹1000
C14  Domestic    900   980   80     ₹240
C15  Commercial  700   820   120    ₹760
```

JUSTIFICATION:

This version simplifies billing by using fixed per-unit rates for each customer type. It improves readability and demonstrates an alternative approach to energy charge calculation.

TASK 3:

Create a Python program using user-defined functions to calculate

energy charges and fixed charges with clear comments.

```
customers = [
    ("C1", "Domestic", 120, 180),
    ("C2", "Domestic", 340, 420),
    ("C3", "Domestic", 500, 650),
    ("C4", "Domestic", 50, 100),
    ("C5", "Domestic", 200, 260),
    ("C6", "Commercial", 1000, 1120),
    ("C7", "Commercial", 450, 600),
    ("C8", "Commercial", 800, 880),
    ("C9", "Commercial", 300, 420),
    ("C10", "Commercial", 1500, 1650),
    ("C11", "Domestic", 600, 690),
    ("C12", "Domestic", 50, 170),
    ("C13", "Commercial", 200, 350),
    ("C14", "Domestic", 900, 980),
    ("C15", "Commercial", 700, 820)
]

# ------------------------------
# Rate Tables
# ------------------------------
energy_rates = {
    "Domestic": (3, 5),
    "Commercial": (6, 8),
    "Industrial": (8, 10)
}
```

```
fixed_charges = {
    "Domestic": 100,
    "Commercial": 250,
    "Industrial": 500
}

# ------------------------------
# Function to Calculate Energy Charges
# ------------------------------
def calculate_energy_charge(cust_type, units):
    """
    Calculates Energy Charges (EC) using formula method
    """
    r1, r2 = energy_rates[cust_type]
    return (min(units, 100) * r1) + (max(units - 100, 0) * r2)

# ------------------------------
# Function to Calculate Fixed Charges
# ------------------------------
def calculate_fixed_charge(cust_type):
    """
    Returns Fixed Charges (FC) based on consumer type
    """
    return fixed_charges[cust_type]
```

```
# ----------------------------
# Billing Output
# ----------------------------
print("Electricity Billing Details- Modular version")
print("-----------------------------------------------------------------
print("Cust  Type        PU    CU    Units   EC (₹)   FC (₹)   Total Bi
print("-----------------------------------------------------------------

for cust_id, cust_type, PU, CU in customers:
    units = CU - PU

    ec = calculate_energy_charge(cust_type, units)
    fc = calculate_fixed_charge(cust_type)
    total_bill = ec + fc

    print(f"{cust_id:<4} {cust_type:<10} {PU:<5} {CU:<5} {units:<7} ₹{
```

```
Electricity Billing Details- Modular version
-----------------------------------------------------------------------
Cust  Type        PU    CU    Units   EC (₹)   FC (₹)   Total Bill (
-----------------------------------------------------------------------
C1   Domestic   120   180   60      ₹180     ₹100     ₹280
C2   Domestic   340   420   80      ₹240     ₹100     ₹340
C3   Domestic   500   650   150     ₹550     ₹100     ₹650
C4   Domestic   50    160   110     ₹350     ₹100     ₹450
C5   Domestic   200   260   60      ₹180     ₹100     ₹280
C6   Commercial 1000  1120  120     ₹760     ₹250     ₹1010
C7   Commercial 450   600   150     ₹1000    ₹250     ₹1250
C8   Commercial 800   880   80      ₹480     ₹250     ₹730
C9   Commercial 300   420   120     ₹760     ₹250     ₹1010
C10  Commercial 1500  1650  150     ₹1000    ₹250     ₹1250
C11  Domestic   600   690   90      ₹270     ₹100     ₹370
C12  Domestic   50    170   120     ₹400     ₹100     ₹500
C13  Commercial 200   350   150     ₹1000    ₹250     ₹1250
C14  Domestic   900   980   80      ₹240     ₹100     ₹340
C15  Commercial 700   820   120     ₹760     ₹250     ₹1010
```

JUSTIFICATION:

This program uses user-defined functions to calculate energy and fixed charges.
Modular design improves reusability, clarity, and maintainability of the billing logic.

TASK 4:

Modify the billing program to include fixed charges, customer charges,

and electricity duty calculated as a percentage of energy charges.

```python
# Electricity Billing System - Task 4
print("Electricity Billing System with Additional Charges")
print("-------------------------------------------------")

# Function to calculate Energy Charges (EC) based on customer type and units cons
def calculate_energy_charges(units_consumed, customer_type):
    """Calculate energy charges using unit slabs"""
    customer_type = customer_type.lower()
    ec = 0
    if customer_type == "domestic":
        if units_consumed <= 100:
            ec = units_consumed * 1.5
        elif units_consumed <= 200:
            ec = (100 * 1.5) + ((units_consumed - 100) * 2.5)
        else:
            ec = (100 * 1.5) + (100 * 2.5) + ((units_consumed - 200) * 4.0)
    elif customer_type == "commercial":
        if units_consumed <= 100:
            ec = units_consumed * 2.5
        elif units_consumed <= 200:
            ec = (100 * 2.5) + ((units_consumed - 100) * 4.0)
        else:
            ec = (100 * 2.5) + (100 * 4.0) + ((units_consumed - 200) * 6.0)
    elif customer_type == "industrial":
        if units_consumed <= 100:
            ec = units_consumed * 3.0
        elif units_consumed <= 200:
            ec = (100 * 3.0) + ((units_consumed - 100) * 5.0)
        else:
            ec = (100 * 3.0) + (100 * 5.0) + ((units_consumed - 200) * 7.0)
```

```python
# Process each customer
for i, cust in enumerate(customers):
    prev_units, curr_units, cust_type = cust
    units_consumed = curr_units - prev_units

    # Calculate charges
    ec = calculate_energy_charges(units_consumed, cust_type)
    fc = calculate_fixed_charges(cust_type)
    cc = calculate_customer_charges(cust_type)
    ed = calculate_electricity_duty(ec, percentage=5)

    total_bill = ec + fc + cc + ed

    # Print detailed bill
    print(f"\n----- Customer {i+1} Bill -----")
    print(f"Previous Units    : {prev_units}")
    print(f"Current Units     : {curr_units}")
    print(f"Units Consumed    : {units_consumed}")
    print(f"Customer Type     : {cust_type}")
    print(f"Energy Charges    : Rs. {ec:.2f}")
    print(f"Fixed Charges     : Rs. {fc:.2f}")
    print(f"Customer Charges  : Rs. {cc:.2f}")
    print(f"Electricity Duty  : Rs. {ed:.2f}")
    print(f"Total Bill        : Rs. {total_bill:.2f}")
```

OUTPUT:

```
    Electricity Billing System with Additional Charges
...  ------------------------------------------------

    ----- Customer 1 Bill -----
    Previous Units   : 120
    Current Units    : 180
    Units Consumed   : 60
    Customer Type    : Domestic
    Energy Charges   : Rs. 90.00
    Fixed Charges    : Rs. 50.00
    Customer Charges : Rs. 20.00
    Electricity Duty : Rs. 4.50
    Total Bill       : Rs. 164.50

    ----- Customer 2 Bill -----
    Previous Units   : 300
    Current Units    : 450
    Units Consumed   : 150
    Customer Type    : Commercial
    Energy Charges   : Rs. 450.00
    Fixed Charges    : Rs. 100.00
    Customer Charges : Rs. 40.00
    Electricity Duty : Rs. 22.50
    Total Bill       : Rs. 612.50

    ----- Customer 3 Bill -----
    Previous Units   : 50
    Current Units    : 120
    Units Consumed   : 70
    Customer Type    : Domestic
    Energy Charges   : Rs. 105.00
    Fixed Charges    : Rs. 50.00
```

```
    Fixed Charges    : Rs. 50.00
    Customer Charges : Rs. 20.00
...  Electricity Duty : Rs. 5.25
    Total Bill       : Rs. 180.25

    ----- Customer 4 Bill -----
    Previous Units   : 600
    Current Units    : 750
    Units Consumed   : 150
    Customer Type    : Commercial
    Energy Charges   : Rs. 450.00
    Fixed Charges    : Rs. 100.00
    Customer Charges : Rs. 40.00
    Electricity Duty : Rs. 22.50
    Total Bill       : Rs. 612.50

    ----- Customer 5 Bill -----
    Previous Units   : 200
    Current Units    : 280
    Units Consumed   : 80
    Customer Type    : Domestic
    Energy Charges   : Rs. 120.00
    Fixed Charges    : Rs. 50.00
    Customer Charges : Rs. 20.00
    Electricity Duty : Rs. 6.00
    Total Bill       : Rs. 196.00
```

JUSTIFICATION:

This code extends modular design by adding customer charges and electricity duty. Each charge is calculated separately to ensure accuracy and transparency in billing.

TASK 5:

Develop the final electricity billing program to calculate total bill

and display all charges clearly with a brief output analysis.

```python
# Final Electricity Bill Generation
print("Final Electricity Bill Generator")
print("--------------------------------")

# Function to calculate Energy Charges (EC)
def calculate_energy_charges(units_consumed, customer_type):
    customer_type = customer_type.lower()
    ec = 0
    if customer_type == "domestic":
        if units_consumed <= 100:
            ec = units_consumed * 1.5
        elif units_consumed <= 200:
            ec = (100 * 1.5) + ((units_consumed - 100) * 2.5)
        else:
            ec = (100 * 1.5) + (100 * 2.5) + ((units_consumed - 200) * 4.0)
    elif customer_type == "commercial":
        if units_consumed <= 100:
            ec = units_consumed * 2.5
        elif units_consumed <= 200:
            ec = (100 * 2.5) + ((units_consumed - 100) * 4.0)
        else:
            ec = (100 * 2.5) + (100 * 4.0) + ((units_consumed - 200) * 6.0)
    elif customer_type == "industrial":
        if units_consumed <= 100:
            ec = units_consumed * 3.0
        elif units_consumed <= 200:
            ec = (100 * 3.0) + ((units_consumed - 100) * 5.0)
        else:
            ec = (100 * 3.0) + (100 * 5.0) + ((units_consumed - 200) * 7.0)
    else:
        ec = 0
    return ec

# Function to calculate Fixed Charges (FC)
def calculate_fixed_charges(customer_type):
    charges = {"domestic": 50, "commercial": 100, "industrial": 150}
    return charges.get(customer_type.lower(), 0)

# Function to calculate Customer Charges (CC)
def calculate_customer_charges(customer_type):
    charges = {"domestic": 20, "commercial": 40, "industrial": 60}
    return charges.get(customer_type.lower(), 0)

# Function to calculate Electricity Duty (ED)
def calculate_electricity_duty(ec, percentage=5):
    return ec * (percentage / 100)

# Sample data: (Previous Units, Current Units, Customer Type)
customers = [
    (120, 180, "Domestic"),
    (300, 450, "Commercial"),
    (50, 120, "Domestic"),
    (600, 750, "Commercial"),
    (200, 280, "Domestic")
]
```

```
# Generate final bills
for i, cust in enumerate(customers):
    prev_units, curr_units, cust_type = cust
    units_consumed = curr_units - prev_units

    # Calculate charges
    ec = calculate_energy_charges(units_consumed, cust_type)
    fc = calculate_fixed_charges(cust_type)
    cc = calculate_customer_charges(cust_type)
    ed = calculate_electricity_duty(ec, 5)
    total_bill = ec + fc + cc + ed

    # Display neatly formatted bill
    print(f"\n----- Customer {i+1} Final Bill -----")
    print(f"Previous Units    : {prev_units}")
    print(f"Current Units     : {curr_units}")
    print(f"Units Consumed    : {units_consumed}")
    print(f"Customer Type     : {cust_type}")
    print(f"Energy Charges    : Rs. {ec:.2f}")
    print(f"Fixed Charges     : Rs. {fc:.2f}")
    print(f"Customer Charges  : Rs. {cc:.2f}")
    print(f"Electricity Duty  : Rs. {ed:.2f}")
    print(f"Total Bill        : Rs. {total_bill:.2f}")

# Short analysis paragraph
print("\nAnalysis:")
print("The program accurately calculates all components of the electricity bill, including Energy Charges, Fixed Charges, Customer Charges, and Electricity Duty. The
```

OUTPUT

```
Final Electricity Bill Generator
---------------------------------

----- Customer 1 Final Bill -----
Previous Units    : 120
Current Units     : 180
Units Consumed    : 60
Customer Type     : Domestic
Energy Charges    : Rs. 90.00
Fixed Charges     : Rs. 50.00
Customer Charges  : Rs. 20.00
Electricity Duty  : Rs. 4.50
Total Bill        : Rs. 164.50

----- Customer 2 Final Bill -----
Previous Units    : 300
Current Units     : 450
Units Consumed    : 150
Customer Type     : Commercial
Energy Charges    : Rs. 450.00
Fixed Charges     : Rs. 100.00
Customer Charges  : Rs. 40.00
Electricity Duty  : Rs. 22.50
Total Bill        : Rs. 612.50

----- Customer 3 Final Bill -----
Previous Units    : 50
Current Units     : 120
Units Consumed    : 70
Customer Type     : Domestic
Energy Charges    : Rs. 105.00
Fixed Charges     : Rs. 50.00
Customer Charges  : Rs. 20.00
Electricity Duty  : Rs. 5.25
Total Bill        : Rs. 180.25

----- Customer 4 Final Bill -----
Previous Units    : 600
Current Units     : 750
Units Consumed    : 150
Customer Type     : Commercial
Energy Charges    : Rs. 450.00
Fixed Charges     : Rs. 100.00
Customer Charges  : Rs. 40.00
Electricity Duty  : Rs. 22.50
Total Bill        : Rs. 612.50

----- Customer 5 Final Bill -----
Previous Units    : 200
Current Units     : 280
Units Consumed    : 80
Customer Type     : Domestic
Energy Charges    : Rs. 120.00
Fixed Charges     : Rs. 50.00
Customer Charges  : Rs. 20.00
Electricity Duty  : Rs. 6.00
Total Bill        : Rs. 196.00

Analysis:
```

JUSTIFICATION

This final version integrates all billing components to generate a complete electricity bill. It produces a neatly formatted output and demonstrates real-world applicability with clear analysis.