

ASSIGNMENT – 3.3

Name: K.Rishitha

Batch: 23

HT No: 2303A51680

Task 1: AI-Generated Logic for Reading Consumer Details

Scenario

An electricity billing system must collect accurate consumer data.

Task Description

Use an AI tool (GitHub Copilot / Gemini) to generate a Python program that:

- Reads:
 - Previous Units (PU)
 - Current Units (CU)
 - Type of Customer
- Calculates units consumed
- Implements logic directly in the main program (no functions)

Expected Output

- Correct input reading
- Units consumed calculation
- Screenshot showing AI-generated code
- Sample input and output

The screenshot shows a code editor window with multiple tabs and a command-line interface at the bottom. The tabs include 'Untitled-1', 'Untitled-2', 'Untitled-3', and 'electric.py'. The command-line shows the path 'C:\Users\souky\Downloads> electric.py>...'. The code itself is a Python function named 'calculate_bill' that takes three parameters: 'previous_units', 'current_units', and 'customer_type'. It calculates the difference between current and previous units, raises a ValueError if it's negative. It then uses a series of if-elif statements to determine a rate based on the customer type ('residential' or 'commercial') and the units consumed. For residential customers, rates are 0.5 for up to 100 units, 0.75 for up to 300 units, and 1.0 for more than 300 units. For commercial customers, rates are 0.75 for up to 100 units, 1.0 for up to 300 units, and 1.5 for more than 300 units.

```
1 # Write a python program for an electricity billing system that reads previous units and current units and type of customer it should calculate bill
2 def calculate_bill(previous_units, current_units, customer_type):
3     units_consumed = current_units - previous_units
4     if units_consumed < 0:
5         raise ValueError("Current units must be greater than or equal to previous units.")
6     if customer_type.lower() == 'residential':
7         if units_consumed <= 100:
8             rate = 0.5
9         elif units_consumed <= 300:
10            rate = 0.75
11        else:
12            rate = 1.0
13    elif customer_type.lower() == 'commercial':
14        if units_consumed <= 100:
15            rate = 0.75
16        elif units_consumed <= 300:
17            rate = 1.0
18        else:
19            rate = 1.5
20    else:
```

```
2 def calculate_bill(previous_units, current_units, customer_type):
3     if customer_type.lower() == "commercial":
4         if units_consumed <= 100:
5             rate = 0.75
6         elif units_consumed <= 300:
7             rate = 1.0
8         else:
9             rate = 1.5
10    else:
11        raise ValueError("Invalid customer type. Please enter 'residential' or 'commercial'.")
12    return units_consumed, units_consumed * rate
13
14 if __name__ == "__main__":
15     try:
16         previous_units = float(input("Enter previous meter reading (units): "))
17         current_units = float(input("Enter current meter reading (units): "))
18         customer_type = input("Enter customer type (residential/commercial): ")
19         units_consumed, total_bill = calculate_bill(previous_units, current_units, customer_type)
20         print(f"Units Consumed: {units_consumed}")
21         print(f"Total Bill Amount: ${total_bill:.2f}")
22     except ValueError as e:
23         print(f"Error: {e}")
24
25
26
27
28
29
30
31
32
```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\souky> & C:/Users/souky/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/souky/Downloads/electric.py
Enter previous meter reading (units): 5000
Enter current meter reading (units): 6500
Enter customer type (residential/commercial): residential
Units Consumed: 1500.0
Total Bill Amount: \$1500.00
○ PS C:\Users\souky>

Task 2: Energy Charges Calculation Based on Units Consumed

Scenario

Energy charges depend on the number of units consumed and customer type.

Task Description

Review the AI-generated code from Task 1 and extend it to:

- Calculate Energy Charges (EC)
- Use conditional statements based on:
 - Domestic
 - Commercial
 - Industrial consumers
- Improve readability using AI prompts such as:
 - “Simplify energy charge calculation logic”
 - “Optimize conditional statements”

Expected Output

- Correct EC calculation
- Clear conditional logic
- Original and improved versions

```

ing reversal code and Untitled-2 ● # Give string reversal code with using f Untitled-3 ● electric.py Untitled-6.py X # Analyze the code with functions and ...
C: > Users > souky > Downloads > Untitled-6.py > ...
1 #In above code energy charges depend on number of units consumed and customer type .now it should calculate energy charge on domes
2 def calculate_bill(previous_reading, current_reading, customer_type):
3     units_consumed = current_reading - previous_reading
4     if units_consumed < 0:
5         raise ValueError("Current unit reading must be greater than or equal to previous unit reading.")
6     if customer_type.lower() == "residential":
7         if units_consumed <= 100:
8             rate = 0.5
9         elif units_consumed <= 300:
10            rate = 0.75
11        else:
12            rate = 1.0
13    elif customer_type.lower() == "commercial":
14        if units_consumed <= 100:
15            rate = 0.75
16        elif units_consumed <= 300:
17            rate = 1.0
18        else:
19            rate = 1.5
20    elif customer_type.lower() == "industrial":
21        if units_consumed <= 100:
22            rate = 1.0

```

```

... ring reversal code and Untitled-2 ● # Give string reversal code with using f Untitled-3 ● electric.py Untitled-6.py X # Analyze the code with functions and ...
C: > Users > souky > Downloads > Untitled-6.py > ...
2 def calculate_bill(previous_reading, current_reading, customer_type):
3     else:
4         rate = 1.5
5     elif customer_type.lower() == "industrial":
6         if units_consumed <= 100:
7             rate = 1.0
8         elif units_consumed <= 300:
9             rate = 1.5
10        else:
11            rate = 2.0
12    else:
13        raise ValueError("Invalid customer type")
14    return units_consumed * rate
15 if __name__ == "__main__":
16     try:
17         previous_units = int(input("Enter previous unit reading: "))
18         current_units = int(input("Enter current unit reading: "))
19         customer_type = input("Enter customer type (residential/commercial/industrial): ")
20         total_bill = calculate_bill(previous_units, current_units, customer_type)
21         print(f"Total bill amount: ${total_bill:.2f}")
22     except ValueError as e:
23         print(f"Error: {e}")
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39

```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\souky> & C:/Users/souky/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/souky/Downloads/electric.py
Total Bill Amount: $1500.00
PS C:\Users\souky> & C:/Users/souky/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/souky/Downloads/Untitled-6.py
File "c:/Users/souky/Downloads/Untitled-6.py", line 37

SyntaxError: expected 'except' or 'finally' block
PS C:\Users\souky> & C:/Users/souky/AppData/Local/Python/pythoncore-3.14-64/python.exe c:/Users/souky/Downloads/Untitled-6.py
Enter previous unit reading: 5000
Enter current unit reading: 6500
Enter customer type (residential/commercial/industrial): commercial
Total bill amount: $2250.00
PS C:\Users\souky>

```

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

Scenario

Billing logic must be reusable for multiple consumers.

Task Description

Use AI assistance to generate a Python program that:

- Uses user-defined functions to:
- Calculate Energy Charges
- Calculate Fixed Charges
- Returns calculated values
- Includes meaningful comments

Expected Output

- Function-based Python program
- Correct EC and FC values
- Screenshots of AI-assisted function generation
- Test cases with outputs

```
C: > Users > souky > Downloads > #In above code billing logic must be reu.py > ...
1  #In above code billing logic must be reusable for multiple consumers it should use defined functions to calculate energy charges, f
2  def calculate_energy_charges(units_consumed, customer_type):
3      if customer_type == 'residential':
4          if units_consumed <= 100:
5              rate = 0.5
6          elif units_consumed <= 300:
7              rate = 0.75
8          else:
9              rate = 1.0
10         elif customer_type == 'commercial':
11             if units_consumed <= 100:
12                 rate = 0.75
13             elif units_consumed <= 300:
14                 rate = 1.0
15             else:
16                 rate = 1.5
17         elif customer_type == 'industrial':
18             if units_consumed <= 100:
19                 rate = 1.0
20             elif units_consumed <= 300:
21                 rate = 1.5
C: > Users > souky > Downloads > #In above code billing logic must be reu.py > ...
2  def calculate_energy_charges(units_consumed, customer_type):
3      if units_consumed <= 300:
4          rate = 1.5
5      else:
6          rate = 2.0
7  else:
8      raise ValueError("Invalid customer type")
9  return units_consumed * rate
10 def calculate_fixed_charges(customer_type):
11     if customer_type == 'residential':
12         return 50
13     elif customer_type == 'commercial':
14         return 100
15     elif customer_type == 'industrial':
16         return 150
17     else:
18         raise ValueError("Invalid customer type")
19 def calculate_bill(previous_units, current_units, customer_type):
20     units_consumed = current_units - previous_units
21     if units_consumed < 0:
22         raise ValueError("Current unit reading must be greater than or equal to previous unit reading")
23     energy_charges = calculate_energy_charges(units_consumed, customer_type)
24     fixed_charges = calculate_fixed_charges(customer_type)
25     return energy_charges + fixed_charges
C: > Users > souky > Downloads > #In above code billing logic must be reu.py > ...
35     raise ValueError('Invalid customer type')
36 def calculate_bill(previous_units, current_units, customer_type):
37     units_consumed = current_units - previous_units
38     if units_consumed < 0:
39         raise ValueError("Current unit reading must be greater than or equal to previous unit reading")
40     energy_charges = calculate_energy_charges(units_consumed, customer_type)
41     fixed_charges = calculate_fixed_charges(customer_type)
42     return energy_charges + fixed_charges
43 if __name__ == "__main__":
44     try:
45         previous_units = int(input("Enter previous unit reading: "))
46         current_units = int(input("Enter current unit reading: "))
47         customer_type = input("Enter customer type (residential/commercial/industrial): ")
48         bill_amount = calculate_bill(previous_units, current_units, customer_type)
49         print(f"Total bill amount: ${bill_amount:.2f}")
50     except ValueError as e:
51         print(f"Error: {e}")
```

```

ust be reu.py"
except ValueError as e:
    ^^^^^^
SyntaxError: invalid syntax
● PS C:\Users\souky> & C:/Users/souky/AppData/Local/Python/pythoncore-3.14-64/py
ust be reu.py"
Enter previous unit reading: 5000
Enter current unit reading: 6500
Enter customer type (residential/commercial/industrial): residential
Total bill amount: $1550.00
○ PS C:\Users\souky>

```

Task 4: Calculation of Additional Charges

Scenario

Electricity bills include multiple additional charges.

Task Description

Extend the program to calculate:

- FC – Fixed Charges
- CC – Customer Charges
- ED – Electricity Duty (percentage of EC)

Use AI prompts like:

- “Add electricity duty calculation”
- “Improve billing accuracy”

Expected Output

- Individual charge values printed
- Correct duty calculation
- Well-structured output
- Verified intermediate results

```

using f Untitled-3 ● electric.py Untitled-6.py # In above code billing logic must be reu.py # In above code it should include multip.py ● # A ▶
C: > Users > souky > Downloads > # In above code it should include multip.py > ...
1  # In above code it should include multiple additional charges liked fixedcharge , customercharge , electricityduty
2  def calculate_energy_charges(units_consumed,customer_type):
3      if customer_type == 'residential':
4          if units_consumed <= 100:
5              rate = 0.5
6          elif units_consumed <= 300:
7              rate = 0.75
8          else:
9              rate = 1.0
10     elif customer_type == 'commercial':
11         if units_consumed <= 100:
12             rate = 0.75
13         elif units_consumed <= 300:
14             rate = 1.0
15         else:
16             rate = 1.5
17     elif customer_type == 'industrial':
18         if units_consumed <= 100:
19             rate = 1.0
20         elif units_consumed <= 300:
21             rate = 1.5
22         else:
23             rate = 2.0

```

```

C: > Users > souky > Downloads > ✎ # In above code it should include multip.py > ...
  2     def calculate_energy_charges(units_consumed, customer_type):
  3         rate = 1.5
  4     else:
  5         rate = 2.0
  6     else:
  7         raise ValueError("Invalid customer type")
  8     return units_consumed * rate
  9 def calculate_fixed_charge(customer_type):
 10     if customer_type == 'residential':
 11         return 50.0
 12     elif customer_type == 'commercial':
 13         return 100.0
 14     elif customer_type == 'industrial':
 15         return 150.0
 16     else:
 17         raise ValueError("Invalid customer type")
 18 def calculate_customer_charge(customer_type):
 19     if customer_type == 'residential':
 20         return 20.0
 21     elif customer_type == 'commercial':
 22         return 40.0
 23     elif customer_type == 'industrial':
 24         return 60.0
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66

```

```

C: > Users > souky > Downloads > ✎ # In above code it should include multip.py > ...
  36     def calculate_customer_charge(customer_type):
 37         if customer_type == 'industrial':
 38             return 60.0
 39         else:
 40             raise ValueError("Invalid customer type")
 41     def calculate_electricity_duty(energy_charges):
 42         return energy_charges * 0.05 # 5% electricity duty
 43     def calculate_bill(previous_units, current_units, customer_type):
 44         units_consumed = current_units - previous_units
 45         if units_consumed < 0:
 46             raise ValueError("Current unit reading must be greater than or equal to previous unit reading")
 47         energy_charges = calculate_energy_charges(units_consumed, customer_type)
 48         fixed_charge = calculate_fixed_charge(customer_type)
 49         customer_charge = calculate_customer_charge(customer_type)
 50         electricity_duty = calculate_electricity_duty(energy_charges)
 51         subtotal = energy_charges + fixed_charge + customer_charge
 52         electricity_duty = calculate_electricity_duty(subtotal)
 53         return subtotal + electricity_duty
 54
 55     if __name__ == "__main__":
 56         try:
 57             previous_units = int(input("Enter previous unit reading: "))
 58             current_units = int(input("Enter current unit reading: "))
 59             customer_type = input("Enter customer type (residential/commercial/industrial): ")
 60             bill = calculate_bill(previous_units, current_units, customer_type)
 61             print(f"Total bill: {bill}")
 62         except ValueError as e:
 63             print(f"Error: {e}")
 64
 65
 66

```

```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\souky> & C:/Users/souky/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/souky/Downloads/#In above code billing logic must be re
Enter previous unit reading: 5000
Enter current unit reading: 6500
Enter customer type (residential/commercial/industrial): residential
Total bill amount: $1550.00
PS C:\Users\souky> & C:/Users/souky/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/souky/Downloads/# In above code it should include multi
Enter previous unit reading: 5000
Enter current unit reading: 6500
Enter customer type (residential/commercial/industrial): residential
Total bill: 1648.5
PS C:\Users\souky>

```

Task 5: Final Bill Generation and Output Analysis

Scenario

The final electricity bill must present all values clearly.

Task Description

Develop the final Python application to:

- Calculate total bill:
- Total Bill = EC + FC + CC + ED
- Display:
 - Energy Charges (EC)
 - Fixed Charges (FC)
 - Customer Charges (CC)
 - Electricity Duty (ED)
 - Total Bill Amount
- Analyze the program based on:
 - Accuracy
 - Readability
 - Real-world applicability

Expected Output

- Complete electricity bill output
- Neatly formatted display
- Sample input/output
- Short analysis paragraph

```
C:\> Users > souky > Downloads > ⚡ #In above code it should print energy ch.py > calculate_customer_charge
1   #In above code it should print energy charges , fixed charges , customer charge ,electricity duty ,total bill amount.
2   def calculate_energy_charges(units_consumed,customer_type):
3       if customer_type == 'residential':
4           if units_consumed <= 100:
5               rate = 0.5
6           elif units_consumed <= 300:
7               rate = 0.75
8           else:
9               rate = 1.0
10      elif customer_type == 'commercial':
11          if units_consumed <= 100:
12              rate = 0.75
13          elif units_consumed <= 300:
14              rate = 1.0
15          else:
16              rate = 1.5
17      elif customer_type == 'industrial':
18          if units_consumed <= 100:
19              rate = 1.0
20          elif units_consumed <= 300:
21              rate = 1.5
22          else:
23              rate = 2.0
24      else:
25          raise ValueError("Invalid customer type")
26      return units_consumed * rate
27  def calculate_fixed_charges(customer_type):
28      if customer_type == 'residential':
29          return 50
30      elif customer_type == 'commercial':
31          return 100
```

```

... using f Untitled-3 ● electric.py ● Untitled-6.py ● #In above code billing logic must be reu.py ● # In above code it should include multip.py ● #In above code it should print energy ch.py
C > Users > souky > Downloads > #In above code it should print energy ch.py > calculate_bill
2 def calculate_energy_charges(units_consumed,customer_type):
open
    24     else:
    25         raise ValueError("Invalid customer type")
    26     return units_consumed * rate
    27 def calculate_fixed_charges(customer_type):
    28     if customer_type == 'residential':
    29         return 50
    30     elif customer_type == 'commercial':
    31         return 100
    32     elif customer_type == 'industrial':
    33         return 150
    34     else:
    35         raise ValueError("Invalid customer type")
    36 def calculate_customer_charge(customer_type):
    37     if customer_type == 'residential':
    38         return 20
    39     elif customer_type == 'commercial':
    40         return 50
    41     elif customer_type == 'industrial':
    42         return 100
    43     else:
    44         raise ValueError("Invalid customer type")
    45 def calculate_electricity_duty(bill_amount):
    46     return bill_amount * 0.05 # 5% electricity duty
    47 def calculate_bill(previous_units, current_units, customer_type):
    48     units_consumed = current_units - previous_units
    49     if units_consumed < 0:
    50         raise ValueError("Current unit reading must be greater than or equal to previous unit reading")
    51     energy_charges = calculate_energy_charges(units_consumed, customer_type)
    52     fixed_charges = calculate_fixed_charges(customer_type)
    53     customer_charge = calculate_customer_charge(customer_type)
    54
    55
    56
    57
    58
    59
    60
    61
    62
    63
    64
    65
    66
    67
    68
    69
    70

```



```

... using f Untitled-3 ● electric.py ● Untitled-6.py ● #In above code billing logic must be reu.py ● # In above code it should include multip.py ● #In above code it should print energy ch.py
C > Users > souky > Downloads > #In above code it should print energy ch.py > calculate_bill
44     raise ValueError("Invalid customer type")
45 def calculate_electricity_duty(bill_amount):
46     return bill_amount * 0.05 # 5% electricity duty
def calculate_bill(previous_units, current_units, customer_type):
48     units_consumed = current_units - previous_units
49     if units_consumed < 0:
50         raise ValueError("Current unit reading must be greater than or equal to previous unit reading")
51     energy_charges = calculate_energy_charges(units_consumed, customer_type)
52     fixed_charges = calculate_fixed_charges(customer_type)
53     customer_charge = calculate_customer_charge(customer_type)
54     subtotal = energy_charges + fixed_charges + customer_charge
55     electricity_duty = calculate_electricity_duty(subtotal)
56     total_bill = subtotal + electricity_duty
57     return energy_charges, fixed_charges, customer_charge, electricity_duty, total_bill
58 if __name__ == "__main__":
59     try:
60         previous_units = int(input("Enter previous unit reading: "))
61         current_units = int(input("Enter current unit reading: "))
62         customer_type = input("Enter customer type (residential/commercial/industrial): ")
63         energy_charges, fixed_charges, customer_charge, electricity_duty, total_bill = calculate_bill(previous_units, current_units, customer_type)
64         print(f"Energy Charges: ${energy_charges:.2f}")
65         print(f"Fixed Charges: ${fixed_charges:.2f}")
66         print(f"Customer Charge: ${customer_charge:.2f}")
67         print(f"Electricity Duty: ${electricity_duty:.2f}")
68         print(f"Total Bill Amount: ${total_bill:.2f}")
69     except ValueError as e:
70         print(f"Error: {e}")

```

PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\souky> & C:/Users/souky/AppData/Local/Python/pythoncore-3.14-6

IndentationError: unindent does not match any outer indentation level

- PS C:\Users\souky> & C:/Users/souky/AppData/Local/Python/pythoncore-3.14-6

```

Enter previous unit reading: 5000
Enter current unit reading: 6000
Enter customer type (residential/commercial/industrial): residential
Energy Charges: $1000.00
Fixed Charges: $50.00
Customer Charge: $20.00
Electricity Duty: $53.50
Total Bill Amount: $1123.50

```

- PS C:\Users\souky>