LAB ASSIGNMENT – 3.1

NAME : Neha Shazneen

ROLL NO : 2403A510A1

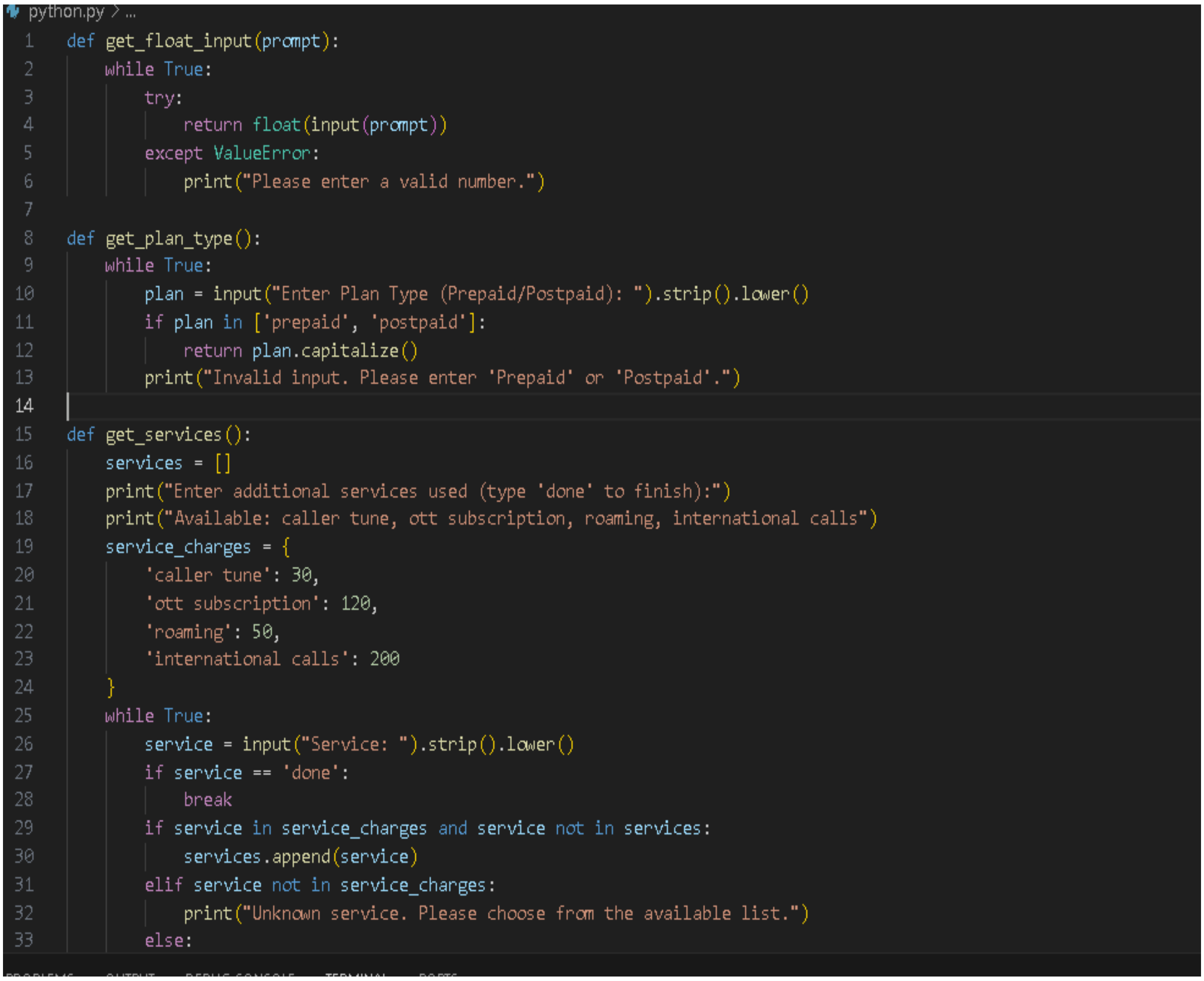
BATCH : 05

SUB : AI ASSISTED CODING

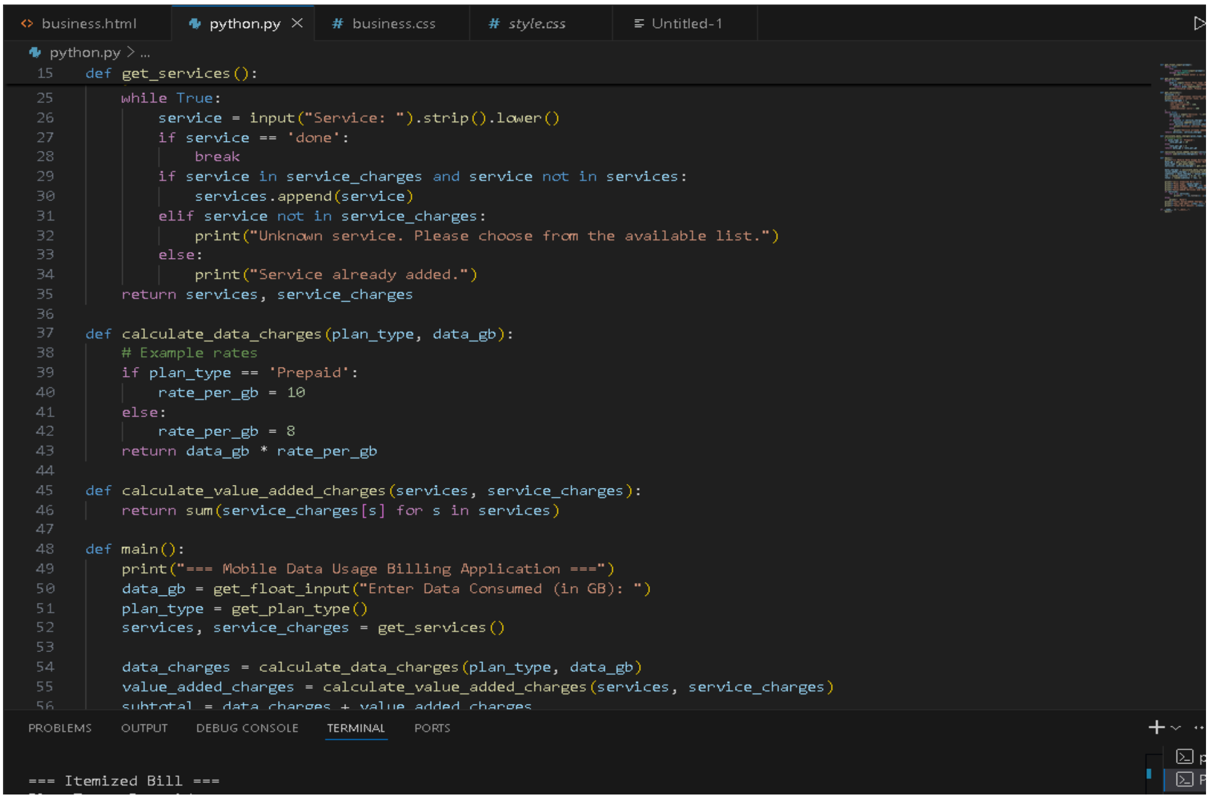
Task – 01:

Prompt:

Write a python code to generate Mobile data usage application bill that includes Read the following from user inputs: Data Consumed (in GB),Plan Type (Prepaid / Postpaid),Additional Services Used (e.g., caller tune, OTTsubscription, etc.). Implement billing logic to calculate:DC (Data Charges) – charges based on dataconsumption, VC (Value-added Charges) – charges for additional services ,Tax – applicable tax on the total bill.

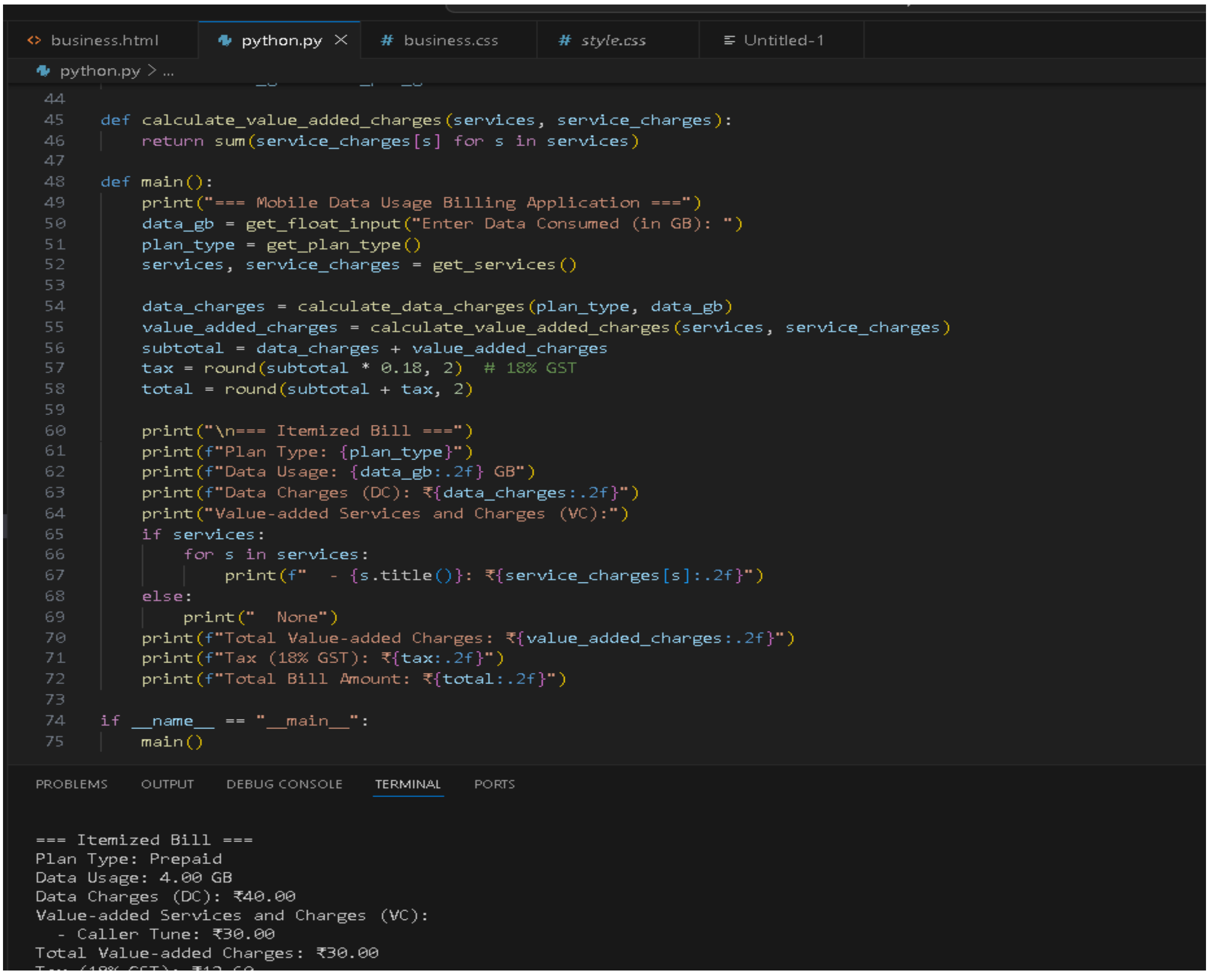


```python
def get_float_input(prompt):
    while True:
        try:
            return float(input(prompt))
        except ValueError:
            print("Please enter a valid number.")

def get_plan_type():
    while True:
        plan = input("Enter Plan Type (Prepaid/Postpaid): ").strip().lower()
        if plan in ['prepaid', 'postpaid']:
            return plan.capitalize()
        print("Invalid input. Please enter 'Prepaid' or 'Postpaid'.")

def get_services():
    services = []
    print("Enter additional services used (type 'done' to finish):")
    print("Available: caller tune, ott subscription, roaming, international calls")
    service_charges = {
        'caller tune': 30,
        'ott subscription': 120,
        'roaming': 50,
        'international calls': 200
    }
    while True:
        service = input("Service: ").strip().lower()
        if service == 'done':
            break
        if service in service_charges and service not in services:
            services.append(service)
        elif service not in service_charges:
            print("Unknown service. Please choose from the available list.")
        else:
```



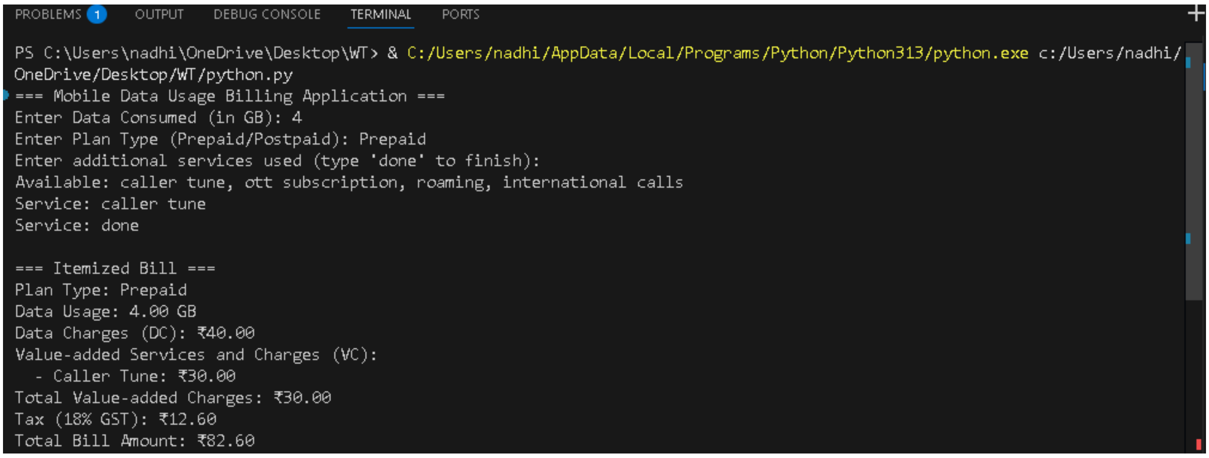
```python
def get_services():
    while True:
        service = input("Service: ").strip().lower()
        if service == 'done':
            break
        if service in service_charges and service not in services:
            services.append(service)
        elif service not in service_charges:
            print("Unknown service. Please choose from the available list.")
        else:
            print("Service already added.")
    return services, service_charges

def calculate_data_charges(plan_type, data_gb):
    # Example rates
    if plan_type == 'Prepaid':
        rate_per_gb = 10
    else:
        rate_per_gb = 8
    return data_gb * rate_per_gb

def calculate_value_added_charges(services, service_charges):
    return sum(service_charges[s] for s in services)

def main():
    print("=== Mobile Data Usage Billing Application ===")
    data_gb = get_float_input("Enter Data Consumed (in GB): ")
    plan_type = get_plan_type()
    services, service_charges = get_services()

    data_charges = calculate_data_charges(plan_type, data_gb)
    value_added_charges = calculate_value_added_charges(services, service_charges)
    subtotal = data_charges + value_added_charges
```

```
=== Itemized Bill ===
```



```python

def calculate_value_added_charges(services, service_charges):
    return sum(service_charges[s] for s in services)

def main():
    print("=== Mobile Data Usage Billing Application ===")
    data_gb = get_float_input("Enter Data Consumed (in GB): ")
    plan_type = get_plan_type()
    services, service_charges = get_services()

    data_charges = calculate_data_charges(plan_type, data_gb)
    value_added_charges = calculate_value_added_charges(services, service_charges)
    subtotal = data_charges + value_added_charges
    tax = round(subtotal * 0.18, 2)  # 18% GST
    total = round(subtotal + tax, 2)

    print("\n=== Itemized Bill ===")
    print(f"Plan Type: {plan_type}")
    print(f"Data Usage: {data_gb:.2f} GB")
    print(f"Data Charges (DC): ₹{data_charges:.2f}")
    print("Value-added Services and Charges (VC):")
    if services:
        for s in services:
            print(f"  - {s.title()}: ₹{service_charges[s]:.2f}")
    else:
        print("  None")
    print(f"Total Value-added Charges: ₹{value_added_charges:.2f}")
    print(f"Tax (18% GST): ₹{tax:.2f}")
    print(f"Total Bill Amount: ₹{total:.2f}")

if __name__ == "__main__":
    main()
```

```
=== Itemized Bill ===
Plan Type: Prepaid
Data Usage: 4.00 GB
Data Charges (DC): ₹40.00
Value-added Services and Charges (VC):
  - Caller Tune: ₹30.00
Total Value-added Charges: ₹30.00
Tax (18% GST): ₹12.60
```



```
PS C:\Users\nadhi\OneDrive\Desktop\WT> & C:/Users/nadhi/AppData/Local/Programs/Python/Python313/python.exe c:/Users/nadhi/
OneDrive/Desktop/WT/python.py
=== Mobile Data Usage Billing Application ===
Enter Data Consumed (in GB): 4
Enter Plan Type (Prepaid/Postpaid): Prepaid
Enter additional services used (type 'done' to finish):
Available: caller tune, ott subscription, roaming, international calls
Service: caller tune
Service: done

=== Itemized Bill ===
Plan Type: Prepaid
Data Usage: 4.00 GB
Data Charges (DC): ₹40.00
Value-added Services and Charges (VC):
  - Caller Tune: ₹30.00
Total Value-added Charges: ₹30.00
Tax (18% GST): ₹12.60
Total Bill Amount: ₹82.60
```

Explanation:

1. The code is a console-based mobile data usage billing application.
2. It prompts the user to enter data consumed (in GB).

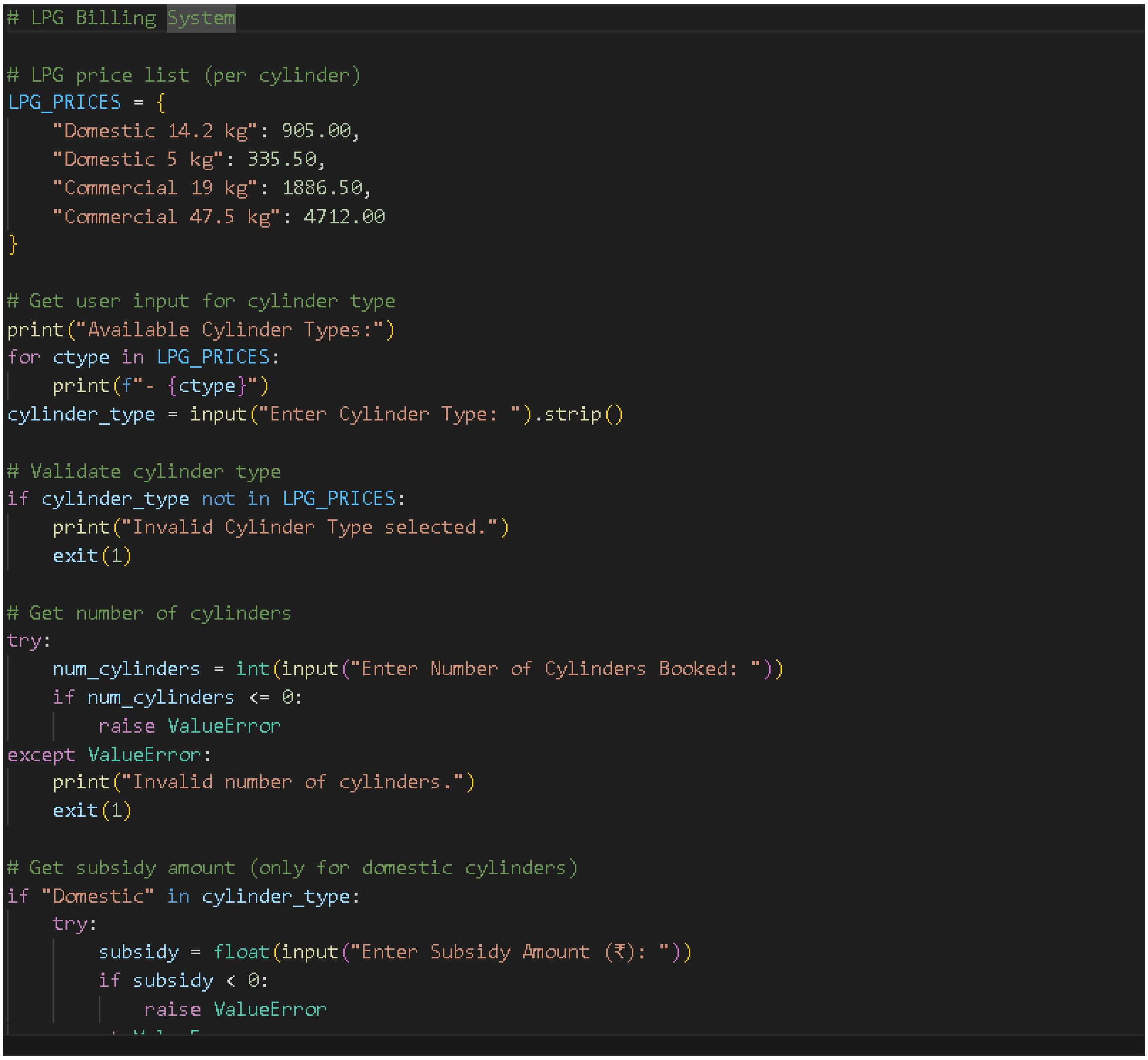
3. The user selects a plan type: Prepaid or Postpaid.
4. The user can add extra services (like caller tune, OTT subscription, etc.).
5. Each service has a fixed charge.
6. Data charges are calculated based on plan type and data used.
7. Value-added charges are summed from selected services.
8. Subtotal is calculated by adding data and value-added charges.
9. An 18% GST tax is added to the subtotal.
10. The program prints an itemized bill with all charges and the total amount.

Task -02:

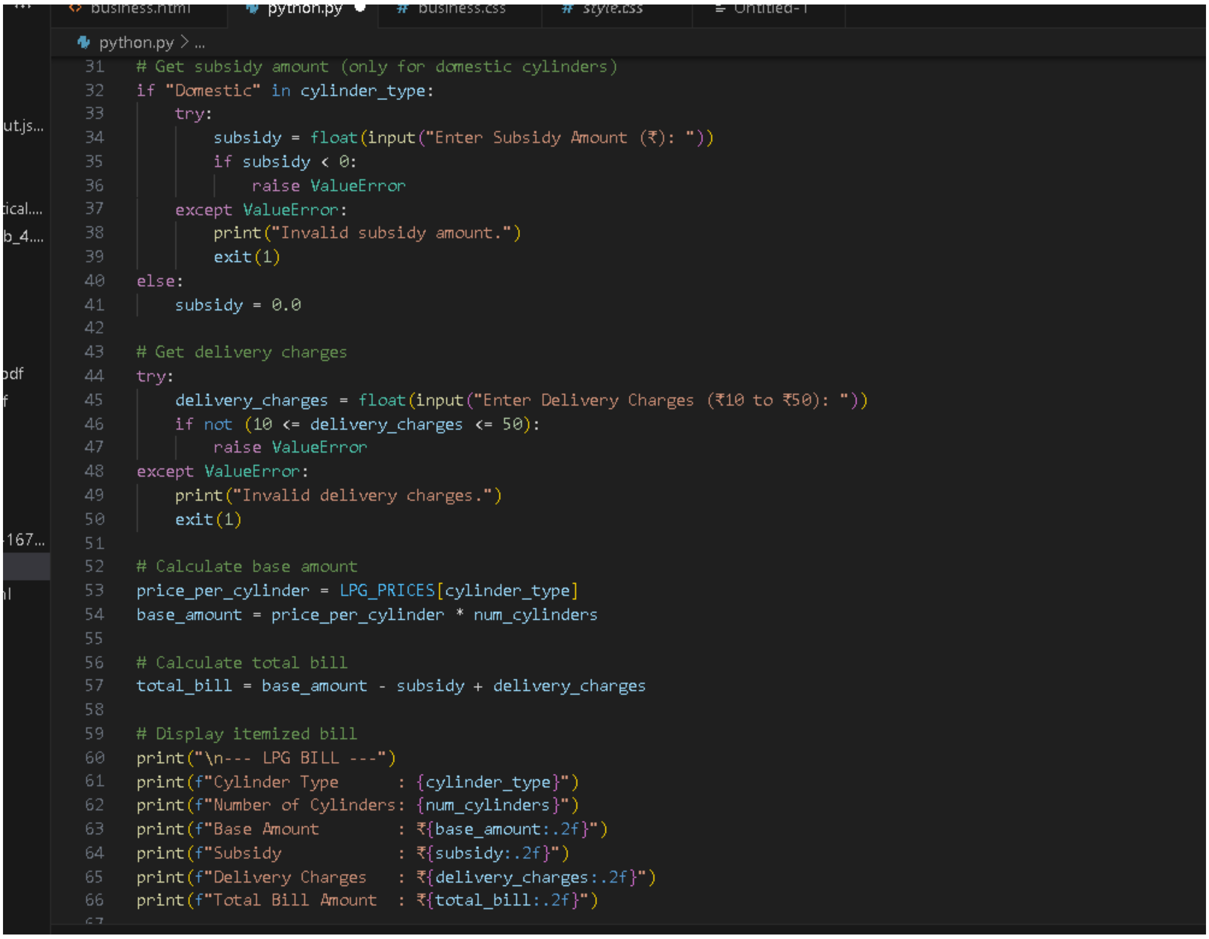
Prompt:

Write a python code to develop a lpg billing system that includes

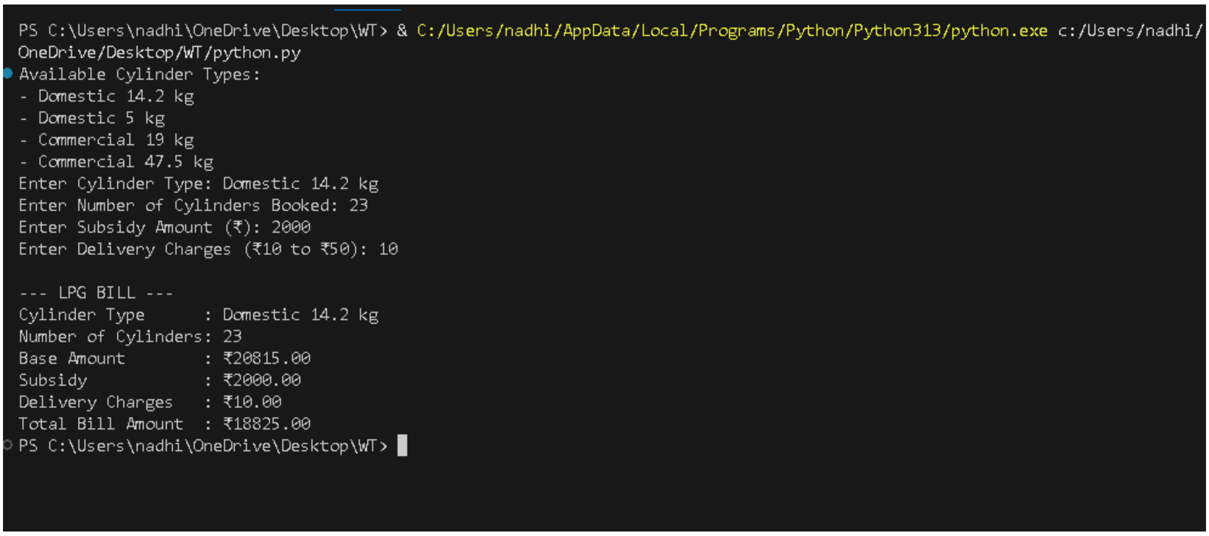
Read the following user inputs:
o Cylinder Type (Domestic 14.2 kg / Domestic 5 kg /
Commercial 19 kg / Commercial 47.5 kg)
o Number of Cylinders Booked
o Subsidy Amount (applicable only for domestic
cylinders)
3. Refer to the given LPG Price List to determine the price per
cylinder:
o Domestic LPG (14.2 kg) → ₹905.00
o Domestic LPG (5 kg) → ₹335.50
o Commercial LPG (19 kg) → ₹1,886.50
o Commercial LPG (47.5 kg) → ₹4,712.00



```python
# LPG Billing System

# LPG price list (per cylinder)
LPG_PRICES = {
    "Domestic 14.2 kg": 905.00,
    "Domestic 5 kg": 335.50,
    "Commercial 19 kg": 1886.50,
    "Commercial 47.5 kg": 4712.00
}

# Get user input for cylinder type
print("Available Cylinder Types:")
for ctype in LPG_PRICES:
    print(f"- {ctype}")
cylinder_type = input("Enter Cylinder Type: ").strip()

# Validate cylinder type
if cylinder_type not in LPG_PRICES:
    print("Invalid Cylinder Type selected.")
    exit(1)

# Get number of cylinders
try:
    num_cylinders = int(input("Enter Number of Cylinders Booked: "))
    if num_cylinders <= 0:
        raise ValueError
except ValueError:
    print("Invalid number of cylinders.")
    exit(1)

# Get subsidy amount (only for domestic cylinders)
if "Domestic" in cylinder_type:
    try:
        subsidy = float(input("Enter Subsidy Amount (₹): "))
        if subsidy < 0:
            raise ValueError
```



```python
# Get subsidy amount (only for domestic cylinders)
if "Domestic" in cylinder_type:
    try:
        subsidy = float(input("Enter Subsidy Amount (₹): "))
        if subsidy < 0:
            raise ValueError
    except ValueError:
        print("Invalid subsidy amount.")
        exit(1)
else:
    subsidy = 0.0

# Get delivery charges
try:
    delivery_charges = float(input("Enter Delivery Charges (₹10 to ₹50): "))
    if not (10 <= delivery_charges <= 50):
        raise ValueError
except ValueError:
    print("Invalid delivery charges.")
    exit(1)

# Calculate base amount
price_per_cylinder = LPG_PRICES[cylinder_type]
base_amount = price_per_cylinder * num_cylinders

# Calculate total bill
total_bill = base_amount - subsidy + delivery_charges

# Display itemized bill
print("\n--- LPG BILL ---")
print(f"Cylinder Type      : {cylinder_type}")
print(f"Number of Cylinders: {num_cylinders}")
print(f"Base Amount        : ₹{base_amount:.2f}")
print(f"Subsidy            : ₹{subsidy:.2f}")
print(f"Delivery Charges   : ₹{delivery_charges:.2f}")
print(f"Total Bill Amount  : ₹{total_bill:.2f}")
```



```
PS C:\Users\nadhi\OneDrive\Desktop\WT> & C:/Users/nadhi/AppData/Local/Programs/Python/Python313/python.exe c:/Users/nadhi/
OneDrive/Desktop/WT/python.py
Available Cylinder Types:
- Domestic 14.2 kg
- Domestic 5 kg
- Commercial 19 kg
- Commercial 47.5 kg
Enter Cylinder Type: Domestic 14.2 kg
Enter Number of Cylinders Booked: 23
Enter Subsidy Amount (₹): 2000
Enter Delivery Charges (₹10 to ₹50): 10

--- LPG BILL ---
Cylinder Type      : Domestic 14.2 kg
Number of Cylinders: 23
Base Amount        : ₹20815.00
Subsidy            : ₹2000.00
Delivery Charges   : ₹10.00
Total Bill Amount  : ₹18825.00
PS C:\Users\nadhi\OneDrive\Desktop\WT>
```

Explanation:

1. The code is an LPG billing system for different cylinder types.

2. It displays available LPG cylinder types and their prices.
3. The user selects a cylinder type and enters the number of cylinders.
4. The code validates the cylinder type and quantity.
5. If the cylinder is domestic, it asks for a subsidy amount.
6. It prompts for delivery charges, ensuring they are between ₹10 and ₹50.
7. The base amount is calculated as price per cylinder times quantity.
8. The total bill is computed as base amount minus subsidy plus delivery charges.
9. The code prints an itemized LPG bill with all details.
10. Input errors (invalid type, quantity, subsidy, or charges) are handled gracefully.