AI ASSISTED CODING

LAB-3.5

Thotapally Pranathi

2303A51733

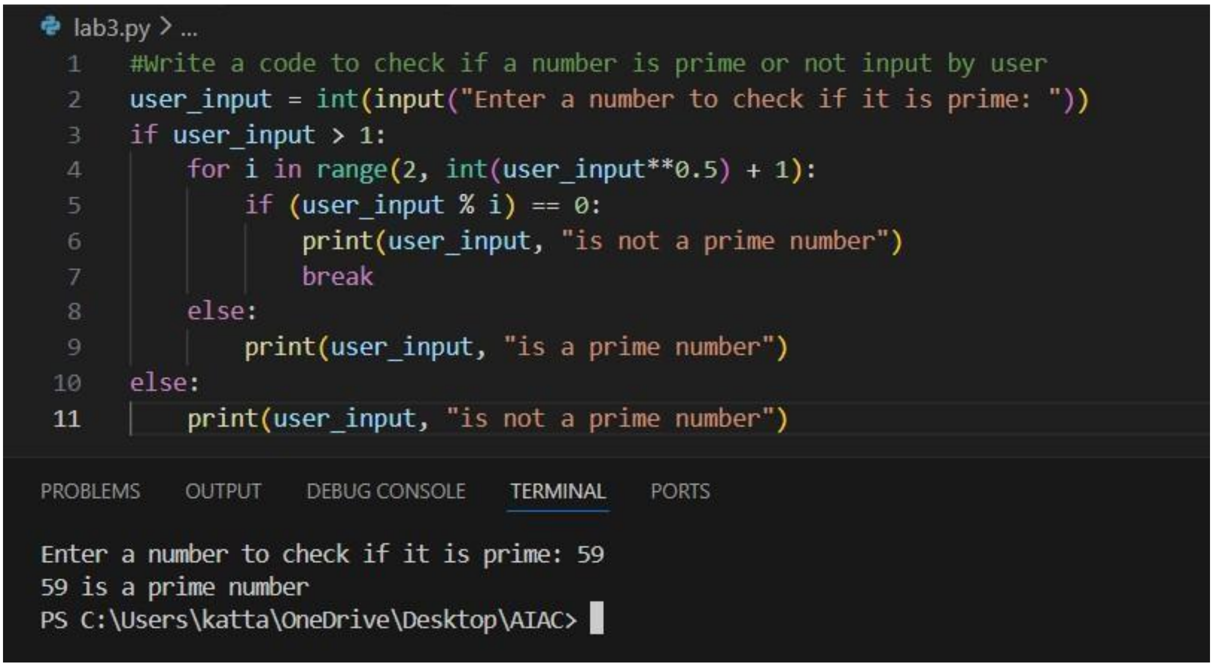
Batch-11

1. Select a simple task: ***"Write a Python function to check if a number is prime."***

Prompt 1:

Write a code to check if a number is prime or not input by user.

Given Code and Output:



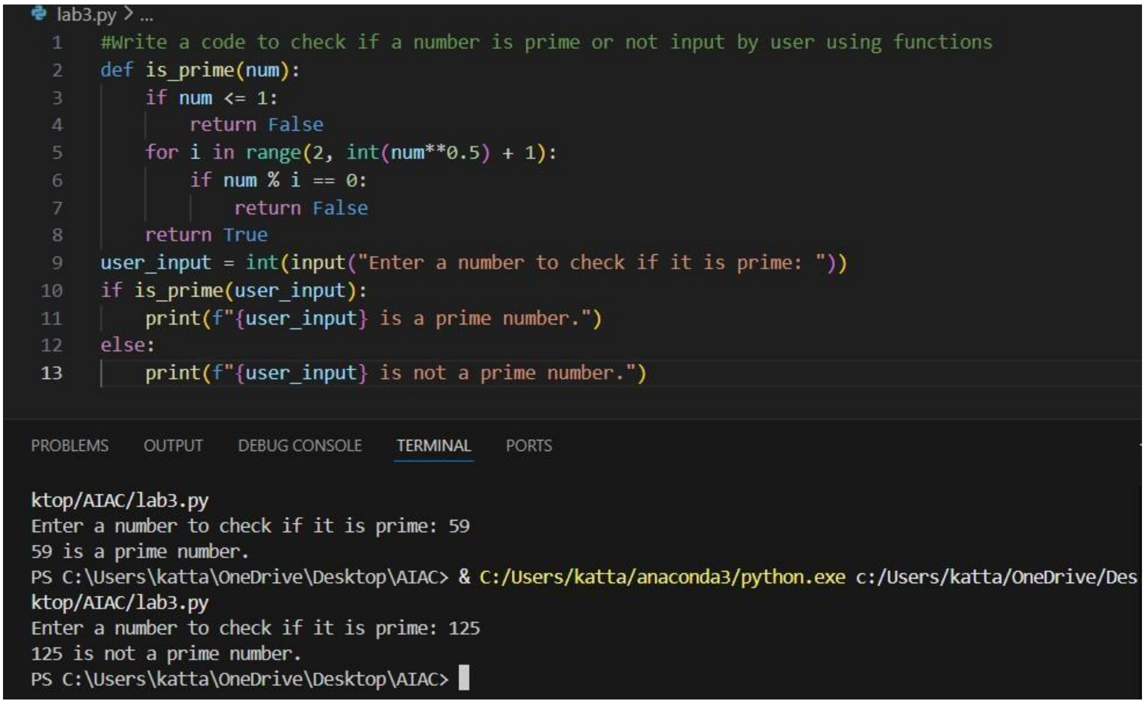
```python
#Write a code to check if a number is prime or not input by user
user_input = int(input("Enter a number to check if it is prime: "))
if user_input > 1:
    for i in range(2, int(user_input**0.5) + 1):
        if (user_input % i) == 0:
            print(user_input, "is not a prime number")
            break
    else:
        print(user_input, "is a prime number")
else:
    print(user_input, "is not a prime number")
```

```
Enter a number to check if it is prime: 59
59 is a prime number
PS C:\Users\katta\OneDrive\Desktop\AIAC>
```

Prompt 2:

Write a code to check if a number is prime or not input by user using functions.

Given Code and Output:

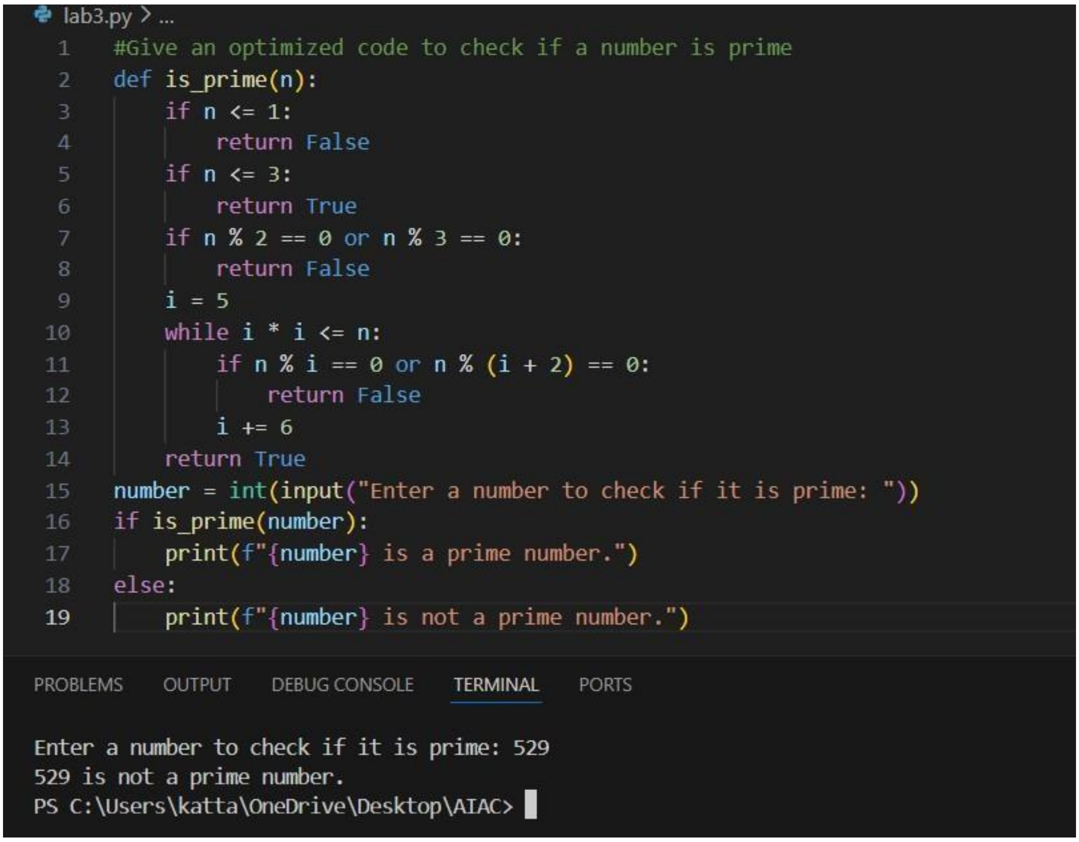


```python
#Write a code to check if a number is prime or not input by user using functions
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True
user_input = int(input("Enter a number to check if it is prime: "))
if is_prime(user_input):
    print(f"{user_input} is a prime number.")
else:
    print(f"{user_input} is not a prime number.")
```

```
ktop/AIAC/lab3.py
Enter a number to check if it is prime: 59
59 is a prime number.
PS C:\Users\katta\OneDrive\Desktop\AIAC> & C:/Users/katta/anaconda3/python.exe c:/Users/katta/OneDrive/Des
ktop/AIAC/lab3.py
Enter a number to check if it is prime: 125
125 is not a prime number.
PS C:\Users\katta\OneDrive\Desktop\AIAC>
```

Prompt 3:

Give an optimized code to check if a number is prime **Given Code and Output:**



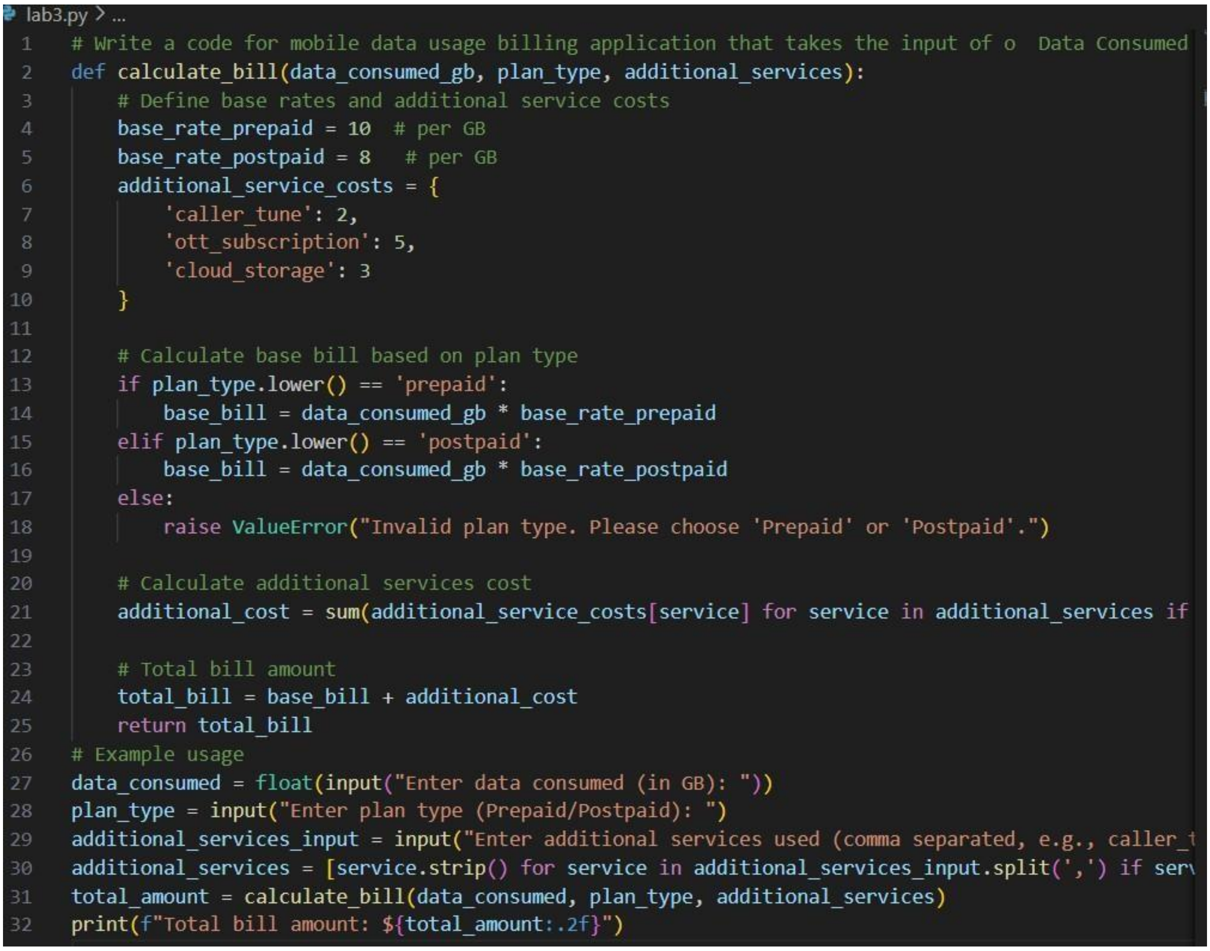
```python
#Give an optimized code to check if a number is prime
def is_prime(n):
    if n <= 1:
        return False
    if n <= 3:
        return True
    if n % 2 == 0 or n % 3 == 0:
        return False
    i = 5
    while i * i <= n:
        if n % i == 0 or n % (i + 2) == 0:
            return False
        i += 6
    return True
number = int(input("Enter a number to check if it is prime: "))
if is_prime(number):
    print(f"{number} is a prime number.")
else:
    print(f"{number} is not a prime number.")
```

```
Enter a number to check if it is prime: 529
529 is not a prime number.
PS C:\Users\katta\OneDrive\Desktop\AIAC>
```

Task: Mobile Data Usage Billing Application (1.0 Marks)

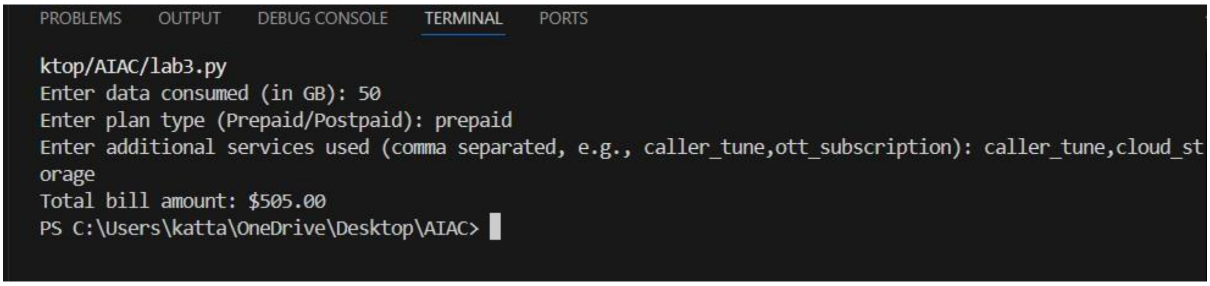
Prompt 1:

Write a code for mobile data usage billing application that takes the input of o Data Consumed (in GB), Plan Type (Prepaid / Postpaid), Additional Services Used (e.g., caller tune, OTT subscription, etc.) and calculate the total bill amount **Code:**



```python
# Write a code for mobile data usage billing application that takes the input of o  Data Consumed
def calculate_bill(data_consumed_gb, plan_type, additional_services):
    # Define base rates and additional service costs
    base_rate_prepaid = 10  # per GB
    base_rate_postpaid = 8   # per GB
    additional_service_costs = {
        'caller_tune': 2,
        'ott_subscription': 5,
        'cloud_storage': 3
    }

    # Calculate base bill based on plan type
    if plan_type.lower() == 'prepaid':
        base_bill = data_consumed_gb * base_rate_prepaid
    elif plan_type.lower() == 'postpaid':
        base_bill = data_consumed_gb * base_rate_postpaid
    else:
        raise ValueError("Invalid plan type. Please choose 'Prepaid' or 'Postpaid'.")

    # Calculate additional services cost
    additional_cost = sum(additional_service_costs[service] for service in additional_services if

    # Total bill amount
    total_bill = base_bill + additional_cost
    return total_bill
# Example usage
data_consumed = float(input("Enter data consumed (in GB): "))
plan_type = input("Enter plan type (Prepaid/Postpaid): ")
additional_services_input = input("Enter additional services used (comma separated, e.g., caller_t
additional_services = [service.strip() for service in additional_services_input.split(',') if serv
total_amount = calculate_bill(data_consumed, plan_type, additional_services)
print(f"Total bill amount: ${total_amount:.2f}")
```

Output:

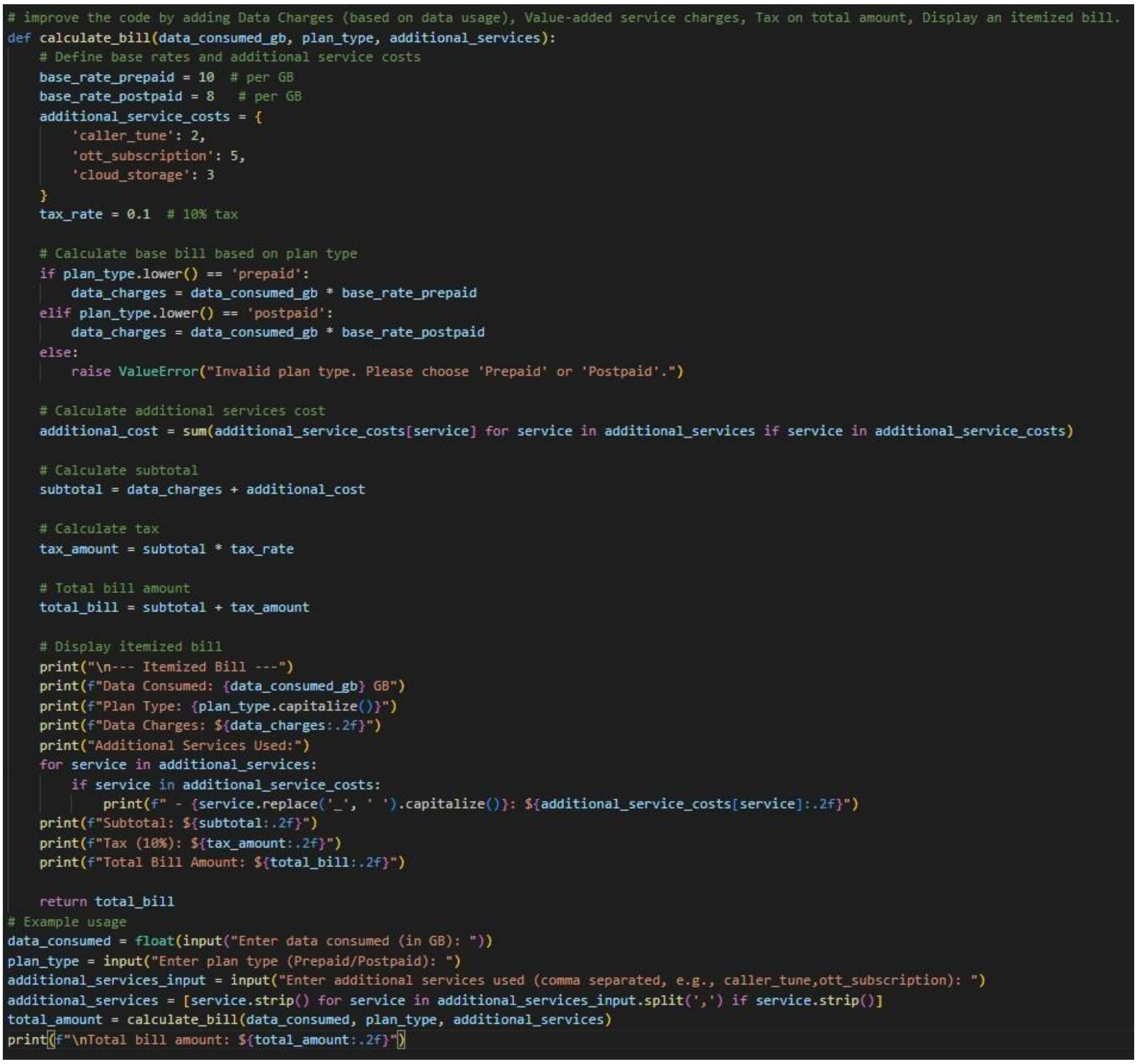


```
ktop/AIAC/lab3.py
Enter data consumed (in GB): 50
Enter plan type (Prepaid/Postpaid): prepaid
Enter additional services used (comma separated, e.g., caller_tune,ott_subscription): caller_tune,cloud_st
orage
Total bill amount: $505.00
PS C:\Users\katta\OneDrive\Desktop\AIAC> █
```

Prompt 2:

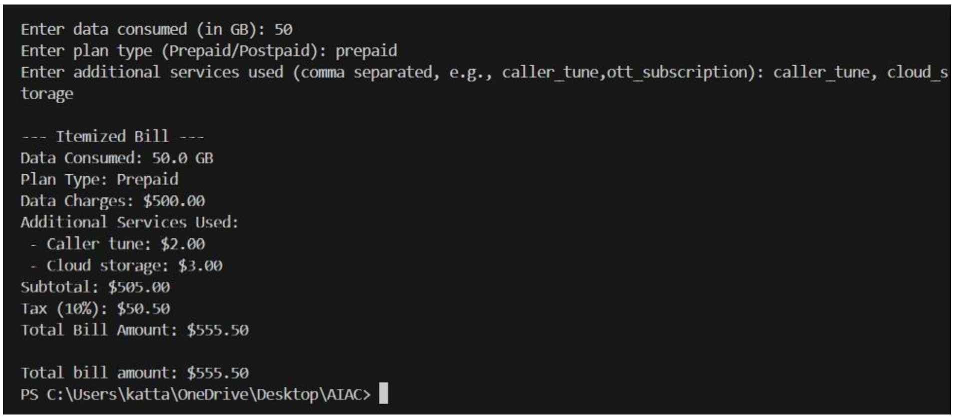
improve the code by adding Data Charges (based on data usage), Value-added service charges, Tax on total amount, Display an itemized bill.

Code:



```python
# improve the code by adding Data Charges (based on data usage), Value-added service charges, Tax on total amount, Display an itemized bill.
def calculate_bill(data_consumed_gb, plan_type, additional_services):
    # Define base rates and additional service costs
    base_rate_prepaid = 10  # per GB
    base_rate_postpaid = 8   # per GB
    additional_service_costs = {
        'caller_tune': 2,
        'ott_subscription': 5,
        'cloud_storage': 3
    }
    tax_rate = 0.1  # 10% tax

    # Calculate base bill based on plan type
    if plan_type.lower() == 'prepaid':
        data_charges = data_consumed_gb * base_rate_prepaid
    elif plan_type.lower() == 'postpaid':
        data_charges = data_consumed_gb * base_rate_postpaid
    else:
        raise ValueError("Invalid plan type. Please choose 'Prepaid' or 'Postpaid'.")

    # Calculate additional services cost
    additional_cost = sum(additional_service_costs[service] for service in additional_services if service in additional_service_costs)

    # Calculate subtotal
    subtotal = data_charges + additional_cost

    # Calculate tax
    tax_amount = subtotal * tax_rate

    # Total bill amount
    total_bill = subtotal + tax_amount

    # Display itemized bill
    print("\n--- Itemized Bill ---")
    print(f"Data Consumed: {data_consumed_gb} GB")
    print(f"Plan Type: {plan_type.capitalize()}")
    print(f"Data Charges: ${data_charges:.2f}")
    print("Additional Services Used:")
    for service in additional_services:
        if service in additional_service_costs:
            print(f" - {service.replace('_', ' ').capitalize()}: ${additional_service_costs[service]:.2f}")
    print(f"Subtotal: ${subtotal:.2f}")
    print(f"Tax (10%): ${tax_amount:.2f}")
    print(f"Total Bill Amount: ${total_bill:.2f}")

    return total_bill
# Example usage
data_consumed = float(input("Enter data consumed (in GB): "))
plan_type = input("Enter plan type (Prepaid/Postpaid): ")
additional_services_input = input("Enter additional services used (comma separated, e.g., caller_tune,ott_subscription): ")
additional_services = [service.strip() for service in additional_services_input.split(',') if service.strip()]
total_amount = calculate_bill(data_consumed, plan_type, additional_services)
print(f"\nTotal bill amount: ${total_amount:.2f}")
```

Output:

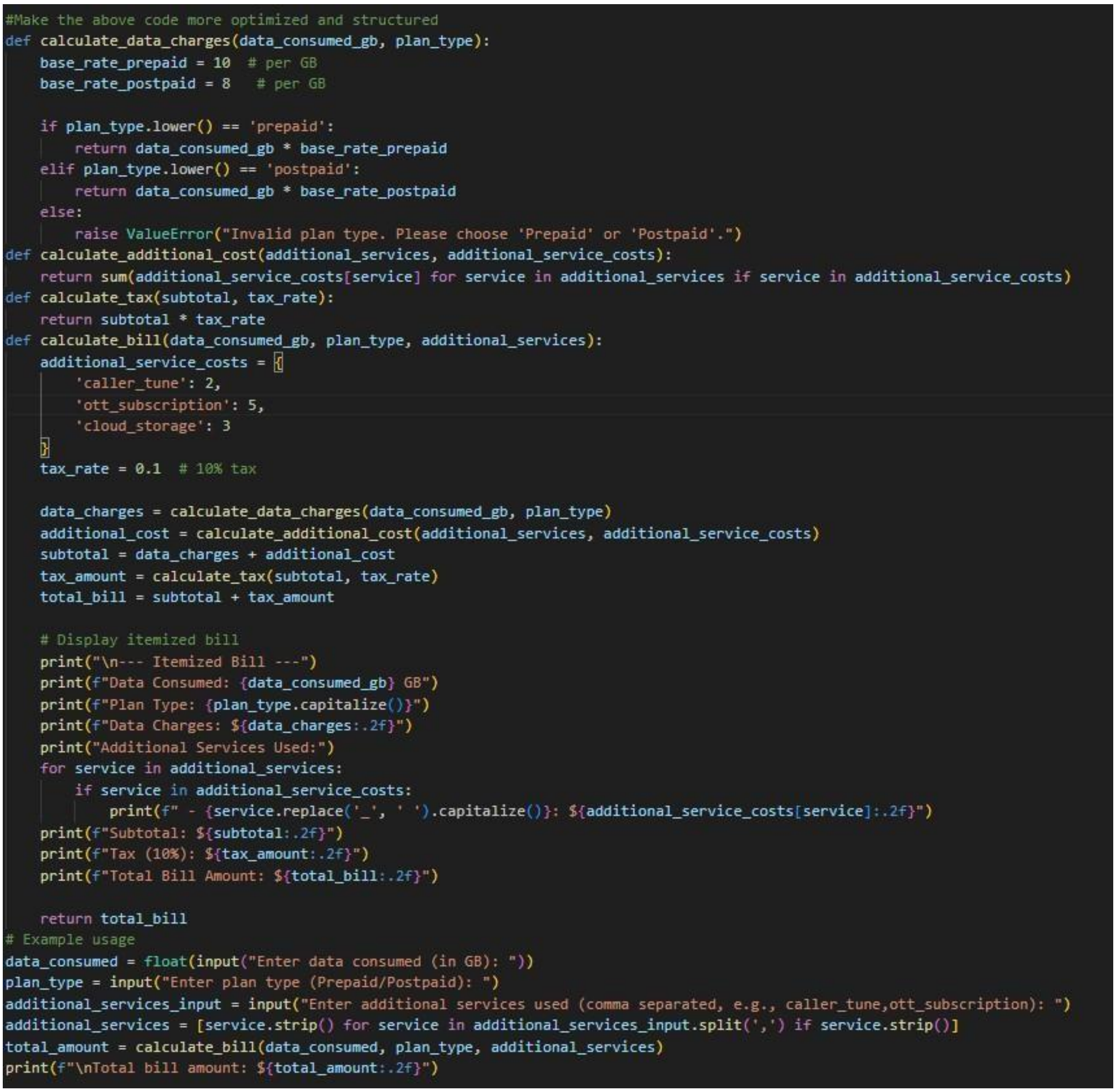


```
Enter data consumed (in GB): 50
Enter plan type (Prepaid/Postpaid): prepaid
Enter additional services used (comma separated, e.g., caller_tune,ott_subscription): caller_tune, cloud_s
torage

--- Itemized Bill ---
Data Consumed: 50.0 GB
Plan Type: Prepaid
Data Charges: $500.00
Additional Services Used:
 - Caller tune: $2.00
 - Cloud storage: $3.00
Subtotal: $505.00
Tax (10%): $50.50
Total Bill Amount: $555.50

Total bill amount: $555.50
PS C:\Users\katta\OneDrive\Desktop\AIAC> 
```

Prompt 3:

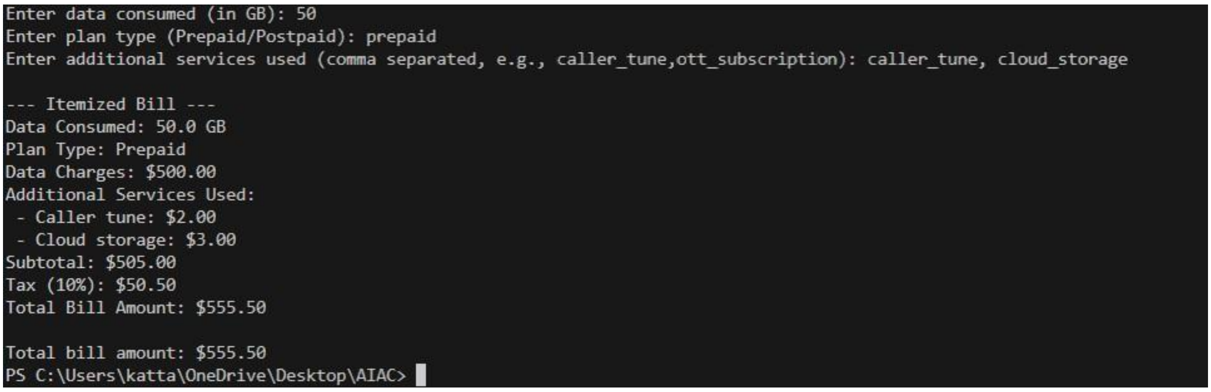
Make the above code more optimized and structured

Code:



```python
#Make the above code more optimized and structured
def calculate_data_charges(data_consumed_gb, plan_type):
    base_rate_prepaid = 10  # per GB
    base_rate_postpaid = 8   # per GB

    if plan_type.lower() == 'prepaid':
        return data_consumed_gb * base_rate_prepaid
    elif plan_type.lower() == 'postpaid':
        return data_consumed_gb * base_rate_postpaid
    else:
        raise ValueError("Invalid plan type. Please choose 'Prepaid' or 'Postpaid'.")
def calculate_additional_cost(additional_services, additional_service_costs):
    return sum(additional_service_costs[service] for service in additional_services if service in additional_service_costs)
def calculate_tax(subtotal, tax_rate):
    return subtotal * tax_rate
def calculate_bill(data_consumed_gb, plan_type, additional_services):
    additional_service_costs = {
        'caller_tune': 2,
        'ott_subscription': 5,
        'cloud_storage': 3
    }
    tax_rate = 0.1  # 10% tax

    data_charges = calculate_data_charges(data_consumed_gb, plan_type)
    additional_cost = calculate_additional_cost(additional_services, additional_service_costs)
    subtotal = data_charges + additional_cost
    tax_amount = calculate_tax(subtotal, tax_rate)
    total_bill = subtotal + tax_amount

    # Display itemized bill
    print("\n--- Itemized Bill ---")
    print(f"Data Consumed: {data_consumed_gb} GB")
    print(f"Plan Type: {plan_type.capitalize()}")
    print(f"Data Charges: ${data_charges:.2f}")
    print("Additional Services Used:")
    for service in additional_services:
        if service in additional_service_costs:
            print(f" - {service.replace('_', ' ').capitalize()}: ${additional_service_costs[service]:.2f}")
    print(f"Subtotal: ${subtotal:.2f}")
    print(f"Tax (10%): ${tax_amount:.2f}")
    print(f"Total Bill Amount: ${total_bill:.2f}")

    return total_bill
# Example usage
data_consumed = float(input("Enter data consumed (in GB): "))
plan_type = input("Enter plan type (Prepaid/Postpaid): ")
additional_services_input = input("Enter additional services used (comma separated, e.g., caller_tune,ott_subscription): ")
additional_services = [service.strip() for service in additional_services_input.split(',') if service.strip()]
total_amount = calculate_bill(data_consumed, plan_type, additional_services)
print(f"\nTotal bill amount: ${total_amount:.2f}")
```

Output:



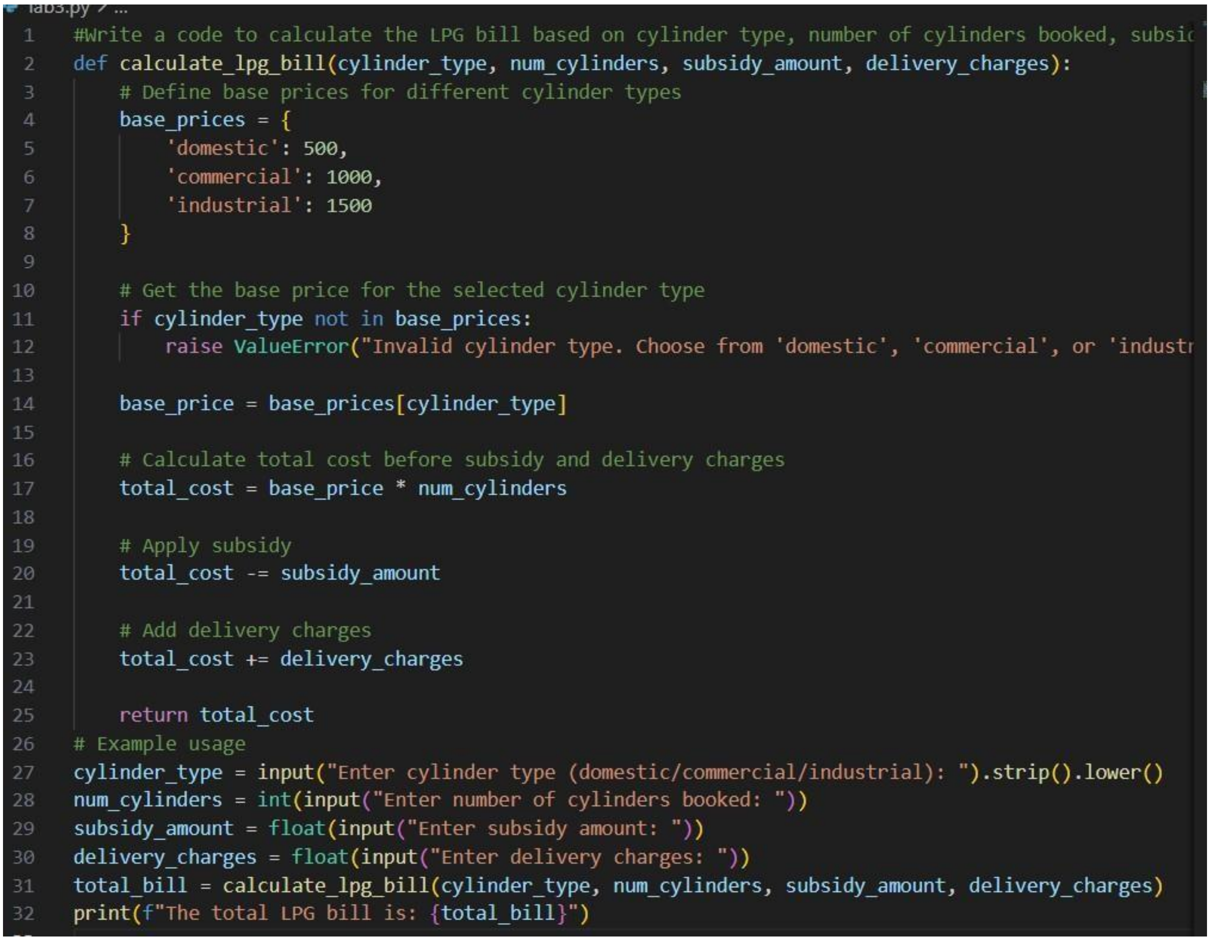
```
Enter data consumed (in GB): 50
Enter plan type (Prepaid/Postpaid): prepaid
Enter additional services used (comma separated, e.g., caller_tune,ott_subscription): caller_tune, cloud_storage

--- Itemized Bill ---
Data Consumed: 50.0 GB
Plan Type: Prepaid
Data Charges: $500.00
Additional Services Used:
 - Caller tune: $2.00
 - Cloud storage: $3.00
Subtotal: $505.00
Tax (10%): $50.50
Total Bill Amount: $555.50

Total bill amount: $555.50
PS C:\Users\katta\OneDrive\Desktop\AIAC> 
```

Task: Develop an LPG Billing System (1.0 Marks) Prompt

1:

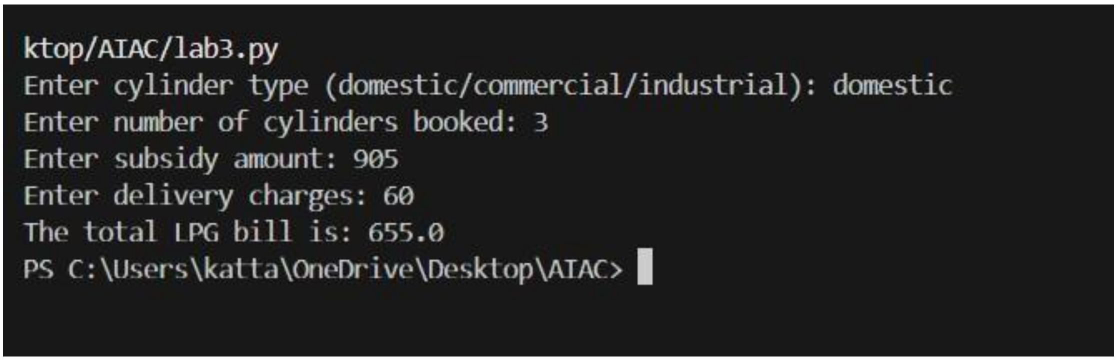
#Write a code to calculate the LPG bill based on cylinder type, number of cylinders booked, subsidy amount, and delivery charges input by user.

Code:



```python
#Write a code to calculate the LPG bill based on cylinder type, number of cylinders booked, subsi
def calculate_lpg_bill(cylinder_type, num_cylinders, subsidy_amount, delivery_charges):
    # Define base prices for different cylinder types
    base_prices = {
        'domestic': 500,
        'commercial': 1000,
        'industrial': 1500
    }

    # Get the base price for the selected cylinder type
    if cylinder_type not in base_prices:
        raise ValueError("Invalid cylinder type. Choose from 'domestic', 'commercial', or 'industr

    base_price = base_prices[cylinder_type]

    # Calculate total cost before subsidy and delivery charges
    total_cost = base_price * num_cylinders

    # Apply subsidy
    total_cost -= subsidy_amount

    # Add delivery charges
    total_cost += delivery_charges

    return total_cost
# Example usage
cylinder_type = input("Enter cylinder type (domestic/commercial/industrial): ").strip().lower()
num_cylinders = int(input("Enter number of cylinders booked: "))
subsidy_amount = float(input("Enter subsidy amount: "))
delivery_charges = float(input("Enter delivery charges: "))
total_bill = calculate_lpg_bill(cylinder_type, num_cylinders, subsidy_amount, delivery_charges)
print(f"The total LPG bill is: {total_bill}")
```

Output:



```
ktop/AIAC/lab3.py
Enter cylinder type (domestic/commercial/industrial): domestic
Enter number of cylinders booked: 3
Enter subsidy amount: 905
Enter delivery charges: 60
The total LPG bill is: 655.0
PS C:\Users\katta\OneDrive\Desktop\AIAC>
```

Explanation:

- In all the tasks at first I asked a basic prompt without any requirements, then the response that was given was very basic.
- After specifying what I wanted in the next prompts the response that was given was more optimised and structured.