Batch-06 **id :2403A51105** **Name:Chaithra alluri**

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		**Assignment Type: Lab**	**Academic Year:**2025-2026
Course Coordinator Name	Venkataramana Veeramsetty		
Instructor(s) Name	Dr. V. Venkataramana (Co-ordinator) Dr. T. Sampath Kumar Dr. Pramoda Patro Dr. Brij Kishor Tiwari Dr.J.Ravichander Dr. Mohammand Ali Shaik Dr. Anirodh Kumar Mr. S.Naresh Kumar Dr. RAJESH VELPULA Mr. Kundhan Kumar Ms. Ch.Rajitha Mr. M Prakash Mr. B.Raju Intern 1 (Dharma teja) Intern 2 (Sai Prasad) Intern 3 (Sowmya) NS2 (Mounika)		
Course Code	24CS002PC215	**Course Title**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	Week2 - Monday	**Time(s)**	
Duration	2 Hours	**Applicable to Batches**	24CSBTB01 To 24CSBTB39
Assignment Number:3.1(Present assignment number)/**24**(Total number of assignments)			

Q.No.	Question	*Expected Time to complete*
1	**Lab Experiment: Prompt Engineering – Improving Prompts and Context Management (0.5 marks)** **Objective**	Week2 - Monday

To explore how prompt design and context influence AI-generated outputs and to learn techniques to improve AI responses.

Tools Required

- GitHub Copilot / Google Gemini / ChatGPT
- VS Code / Google Colab
- Internet access

Procedure

1. Select a simple task: *"Write a Python function to check if a number is prime."*
2. Use different prompting strategies to generate the solution:
   a) Zero-Shot – no examples.
   b) One-Shot – one example provided.
   c) Few-Shot – multiple examples provided.
   d) Context-Managed – detailed prompt with constraints and instructions.
3. Record AI responses and refine prompts to improve code quality.
4. Request AI to optimize the logic for efficiency.
5. Compare results and document improvements.

Sample Prompts

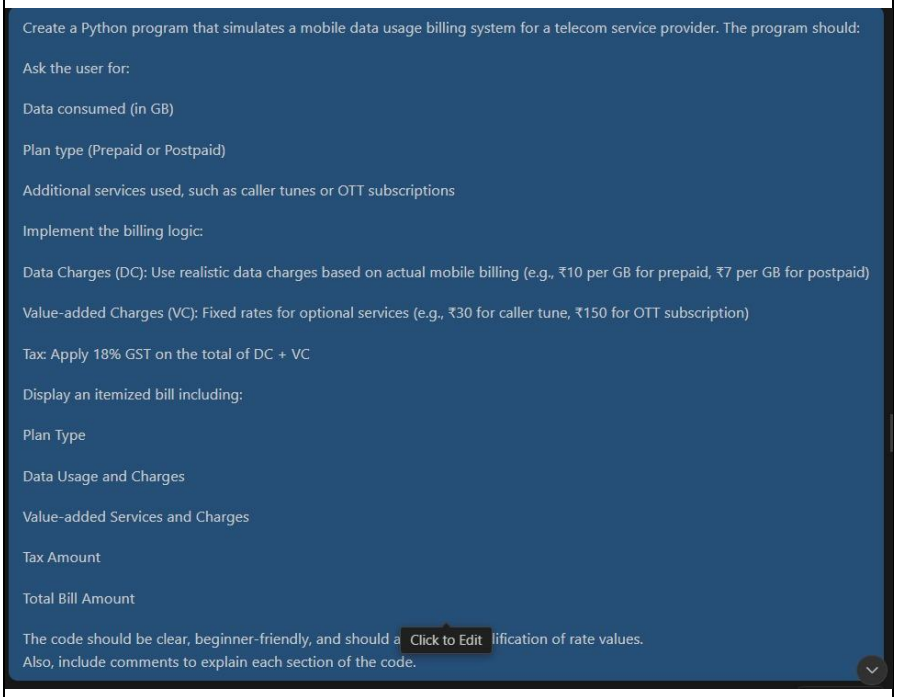
- Zero-Shot:
  Write a Python function to check if a number is prime.
- One-Shot:
  Example: Input: 5 → Output: Prime. Now, write a function to check if a number is prime.
- Few-Shot:
  Example 1: Input: 7 → Output: Prime
  Example 2: Input: 10 → Output: Not Prime
  Example 3: Input: 2 → Output: Prime

	Generate the function accordingly. ● Context-Managed (With Optimization)	
2	**Task: Mobile Data Usage Billing Application (1.0 Marks)** **Objective:** Use Python programming and AI-assisted coding tools to create an application that simulates mobile data billing for a telecom service provider. Instructions 1. Use GitHub Copilot or Google Gemini to assist in writing the program. 2. Read the following inputs from the user: o Data Consumed (in GB) o Plan Type (Prepaid / Postpaid) o Additional Services Used (e.g., caller tune, OTT subscription, etc.) 3. Implement billing logic to calculate: o DC (Data Charges) – charges based on data consumption o VC (Value-added Charges) – charges for additional services o Tax – applicable tax on the total bill 4. Display an itemized bill showing: o Plan Type o Data Usage and Charges o Value-added Services and Charges o Tax o Total Bill Amount Requirements ● Students must refer to their actual mobile bill for charge structure (data cost, service fees, taxes) to make the program realistic. ● AI assistance (Copilot/Gemini) must be used to generate and refine the initial code.	Week2 - Monday

Deliverables

- AI prompts used for code generation.
- AI-generated Python code and any optimized version.
- Screenshots of:
    - AI interactions
    - Program execution and output
    - Comparison with the student's actual mobile bill.

PROMPT:



CODE:

```python
# Mobile Data Usage Billing System for a Telecom Service Provider

# Define rates for easy modification
PREPAID_RATE_PER_GB = 10      # ₹ per GB for prepaid
POSTPAID_RATE_PER_GB = 7      # ₹ per GB for postpaid
CALLER_TUNE_CHARGE = 30       # ₹ fixed charge for caller tune
OTT_SUBSCRIPTION_CHARGE = 150 # ₹ fixed charge for OTT subscription
GST_RATE = 0.18               # 18% GST

def get_plan_type():
    """Ask user for plan type and validate input."""
    while True:
        plan = input("Enter plan type (Prepaid/Postpaid): ").strip().lower()
        if plan in ['prepaid', 'postpaid']:
            return plan
        print("Invalid input. Please enter 'Prepaid' or 'Postpaid'.")

def get_data_usage():
    """Ask user for data consumed in GB and validate input."""
    while True:
        try:
            data = float(input("Enter data consumed (in GB): "))
            if data >= 0:
                return data
            print("Data usage cannot be negative.")
        except ValueError:
            print("Invalid input. Please enter a number.")
```

```python
def get_data_usage():
    """Ask user for data consumed in GB and validate input."""
    while True:
        try:
            data = float(input("Enter data consumed (in GB): "))
            if data >= 0:
                return data
            print("Data usage cannot be negative.")
        except ValueError:
            print("Invalid input. Please enter a number.")

def get_value_added_services():
    """Ask user for additional services and return total value-added charges."""
    services = []
    total_vc = 0
    caller_tune = input("Did you use Caller Tune service? (yes/no): ").strip().lower()
    if caller_tune == 'yes':
        services.append("Caller Tune")
        total_vc += CALLER_TUNE_CHARGE
    ott = input("Did you use OTT Subscription? (yes/no): ").strip().lower()
    if ott == 'yes':
        services.append("OTT Subscription")
        total_vc += OTT_SUBSCRIPTION_CHARGE
    return services, total_vc

def calculate_bill(plan, data_usage, value_services, vc):
    """Calculate data charges, tax, and total bill."""
    if plan == 'prepaid':
        dc = data_usage * PREPAID_RATE_PER_GB
```

```
    """Calculate data charges, tax, and total bill."""
    if plan == 'prepaid':
        dc = data_usage * PREPAID_RATE_PER_GB
    else:
        dc = data_usage * POSTPAID_RATE_PER_GB
    subtotal = dc + vc
    tax = subtotal * GST_RATE
    total = subtotal + tax
    return dc, tax, total

def display_bill(plan, data_usage, dc, value_services, vc, tax, total):
    """Display an itemized bill."""
    print("\n--- Itemized Bill ---")
    print(f"Plan Type: {plan.capitalize()}")
    print(f"Data Used: {data_usage} GB")
    print(f"Data Charges: ₹{dc:.2f}")
    if value_services:
        print("Value-added Services:")
        for service in value_services:
            charge = CALLER_TUNE_CHARGE if service == "Caller Tune" else OTT_SUBSCRIPTION_CHARGE
            print(f"  {service}: ₹{charge}")
        print(f"Total Value-added Charges: ₹{vc}")
    else:
        print("Value-added Services: None")
    print(f"GST (18%): ₹{tax:.2f}")
    print(f"Total Bill Amount: ₹{total:.2f}")

# Main program
if __name__ == "__main__":
```

```
if __name__ == "__main__":
    print("Welcome to the Mobile Data Usage Billing System!\n")
    plan = get_plan_type()
    data_usage = get_data_usage()
    value_services, vc = get_value_added_services()
    dc, tax, total = calculate_bill(plan, data_usage, value_services, vc)
    display_bill(plan, data_usage, dc, value_services, vc, tax, total)
```

OUTPUT:

```
Enter plan type (Prepaid/Postpaid): Postpaid
Enter data consumed (in GB): 10
Did you use Caller Tune service? (yes/no): yes
Did you use OTT Subscription? (yes/no): yes

--- Itemized Bill ---
Plan Type: Postpaid
Data Used: 10.0 GB
Data Charges: ₹70.00
Value-added Services:
  Caller Tune: ₹30
  OTT Subscription: ₹150
Total Value-added Charges: ₹180
GST (18%): ₹44.10
Total Bill Amount: ₹294.10
```

	`` Welcome to the Mobile Data Usage Billing System! Enter plan type (Prepaid/Postpaid): Prepaid Enter data consumed (in GB): 5 Did you use Caller Tune service? (yes/no): yes Did you use OTT Subscription? (yes/no): no --- Itemized Bill --- Plan Type: Prepaid Data Used: 5.0 GB Data Charges: ₹50.00 Value-added Services: Caller Tune: ₹30 Total Value-added Charges: ₹30 GST (18%): ₹14.40 Total Bill Amount: ₹94.40 ```	
3	**Task: Develop an LPG Billing System (1.0 Marks)** **Objective** Apply your Python programming skills and utilize AI-assisted coding tools to build an application that calculates the LPG bill based on specified customer inputs and billing parameters. Instructions 1. Use GitHub Copilot or Google Gemini to assist in writing and refining the program. 2. 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 o Delivery Charges (₹10 to ₹50)	Week2 - Monday

4. Implement the billing formula:

Bill Amount = (Price per Cylinder × Quantity) - Subsidy (if applicable) + Delivery Charges

5. Calculate and display an itemized bill including:

- Cylinder Type
- Number of Cylinders
- Base Amount
- Subsidy
- Delivery Charges
- Total Bill Amount

Deliverables

- A report containing:
    - AI prompts used to generate the program
    - AI-generated Python code
    - Line-by-line explanation of the code

PROMPT:

Create a Python program that simulates an LPG Billing System for different types of gas cylinders. The program should:

Ask the user for the following inputs:

Cylinder Type: Choose from:

Domestic 14.2 kg

Domestic 5 kg

Commercial 19 kg

Commercial 47.5 kg

Number of Cylinders Booked

Subsidy Amount (only applicable for domestic cylinders)

Delivery Charges (random or user input, between ₹10 and ₹50)

Use the following fixed prices for each cylinder type:

CODE:

```python
import random

# Fixed prices for each cylinder type (in ₹)
CYLINDER_PRICES = {
    "Domestic 14.2 kg": 1050,
    "Domestic 5 kg": 400,
    "Commercial 19 kg": 1800,
    "Commercial 47.5 kg": 4500
}

def get_cylinder_type():
    """Prompt user to select cylinder type and validate input."""
    print("Select Cylinder Type:")
    for idx, cyl in enumerate(CYLINDER_PRICES.keys(), 1):
        print(f"{idx}. {cyl}")
    while True:
        try:
            choice = int(input("Enter choice (1-4): "))
            if 1 <= choice <= 4:
                return list(CYLINDER_PRICES.keys())[choice - 1]
            print("Invalid choice. Please select a number between 1 and 4.")
        except ValueError:
            print("Invalid input. Please enter a number.")

def get_number_of_cylinders():
    """Prompt user for number of cylinders booked."""
    while True:
        try:
            num = int(input("Enter number of cylinders booked: "))
            if num > 0:
                return num
            print("Number must be greater than zero.")
        except ValueError:
            print("Invalid input. Please enter a number.")

def get_subsidy_amount(cylinder_type):
    """Prompt user for subsidy amount if domestic cylinder."""
    if "Domestic" in cylinder_type:
        while True:
            try:
                subsidy = float(input("Enter subsidy amount (₹): "))
                if subsidy >= 0:
                    return subsidy
                print("Subsidy cannot be negative.")
            except ValueError:
                print("Invalid input. Please enter a number.")
    else:
        return 0.0

def get_delivery_charges():
    """Prompt user for delivery charges or generate randomly."""
    choice = input("Do you want to enter delivery charges? (yes/no): ").strip().lower()
    if choice == "yes":
        while True:
            try:
                charges = float(input("Enter delivery charges (₹10 to ₹50): "))
                if 10 <= charges <= 50:
                    return charges
```

```python
def calculate_bill(cylinder_type, num_cylinders, subsidy, delivery_charges):
    """Calculate total bill."""
    base_price = CYLINDER_PRICES[cylinder_type] * num_cylinders
    total = base_price + delivery_charges - subsidy
    return base_price, delivery_charges, subsidy, total

def display_bill(cylinder_type, num_cylinders, base_price, delivery_charges, subsidy, total):
    """Display itemized LPG bill."""
    print("\n--- LPG Cylinder Bill ---")
    print(f"Cylinder Type: {cylinder_type}")
    print(f"Number of Cylinders: {num_cylinders}")
    print(f"Base Price: ₹{base_price:.2f}")
    print(f"Delivery Charges: ₹{delivery_charges:.2f}")
    if subsidy > 0:
        print(f"Subsidy Amount: -₹{subsidy:.2f}")
    print(f"Total Bill Amount: ₹{total:.2f}")

# Main program
if __name__ == "__main__":
    print("Welcome to the LPG Billing System!\n")
    cylinder_type = get_cylinder_type()
    num_cylinders = get_number_of_cylinders()
    subsidy = get_subsidy_amount(cylinder_type)
    delivery_charges = get_delivery_charges()
    base_price, delivery_charges, subsidy, total = calculate_bill(
        cylinder_type, num_cylinders, subsidy, delivery_charges
    )
    display_bill(cylinder_type, num_cylinders, base_price, delivery_charges, subsidy, total)
```

OUTPUT:

```
Select Cylinder Type:
1. Domestic 14.2 kg
2. Domestic 5 kg
3. Commercial 19 kg
4. Commercial 47.5 kg
Enter choice (1-4): 3
Enter number of cylinders booked: 1
Do you want to enter delivery charges? (yes/no): no

--- LPG Cylinder Bill ---
Cylinder Type: Commercial 19 kg
Number of Cylinders: 1
Base Price: ₹1800.00
Delivery Charges: ₹37.00
Total Bill Amount: ₹1837.00
```

```
Select Cylinder Type:
1. Domestic 14.2 kg
2. Domestic 5 kg
3. Commercial 19 kg
4. Commercial 47.5 kg
Enter choice (1-4): 1
Enter number of cylinders booked: 2
Enter subsidy amount (₹): 100
Do you want to enter delivery charges? (yes/no): yes
Enter delivery charges (₹10 to ₹50): 25

--- LPG Cylinder Bill ---
Cylinder Type: Domestic 14.2 kg
Number of Cylinders: 2
Base Price: ₹2100.00
Delivery Charges: ₹25.00
Subsidy Amount: -₹100.00
Total Bill Amount: ₹2025.00
```