AI ASSISTED CODING

NAME:Dugyala Ashmitha
ROLL NO:2403A510G5
ASSIGNMENT:3.1

Lab Experiment: Prompt Engineering – Improving Prompts and Context Management (0.5 marks)

Objective

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

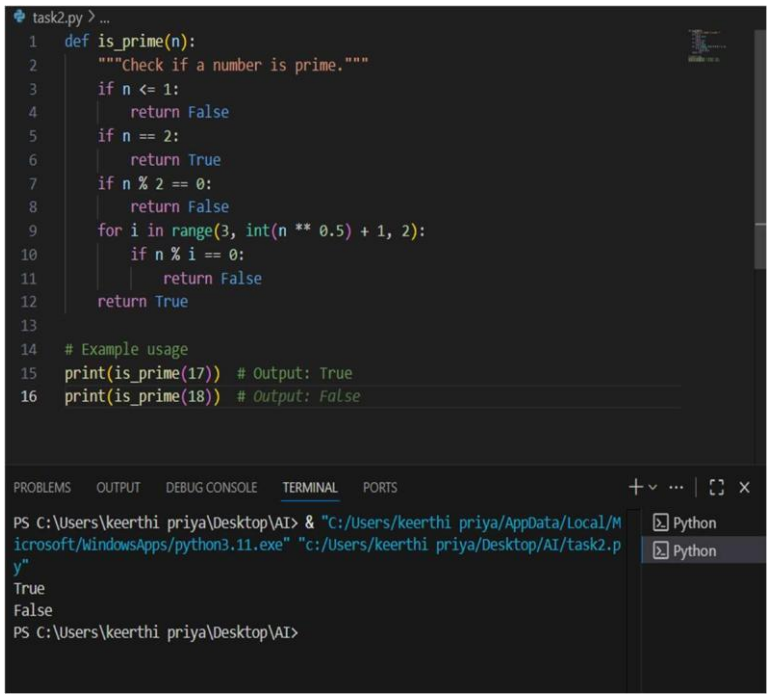
  Example 2: Input: 10 → Output: Not Prime

  Example 3: Input: 2 → Output: Prime

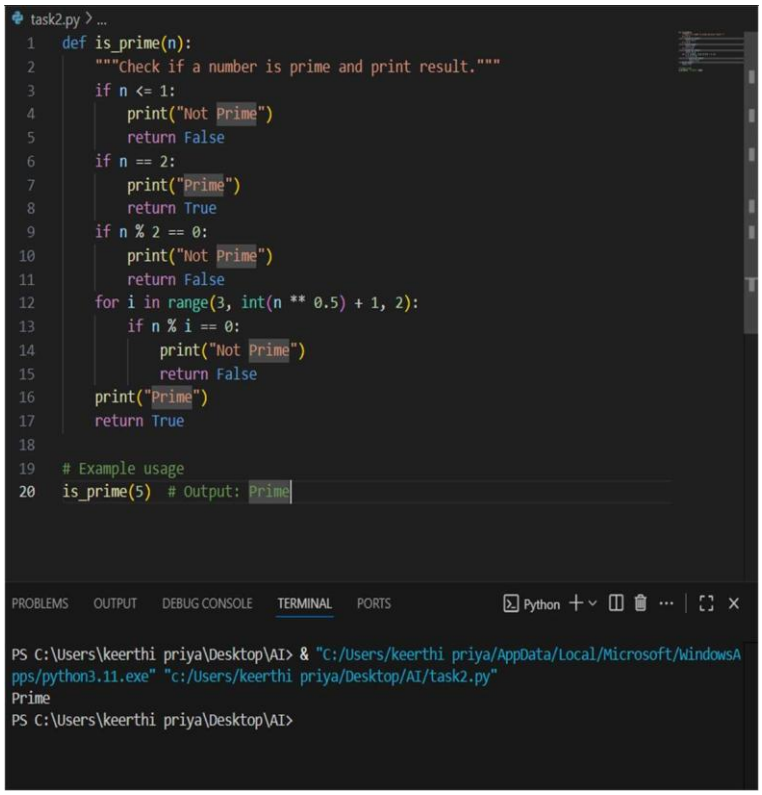
  Generate the function accordingly. ●

Context-Managed (With Optimization)

- ZERO-SHOT:



- ONE-SHOT:

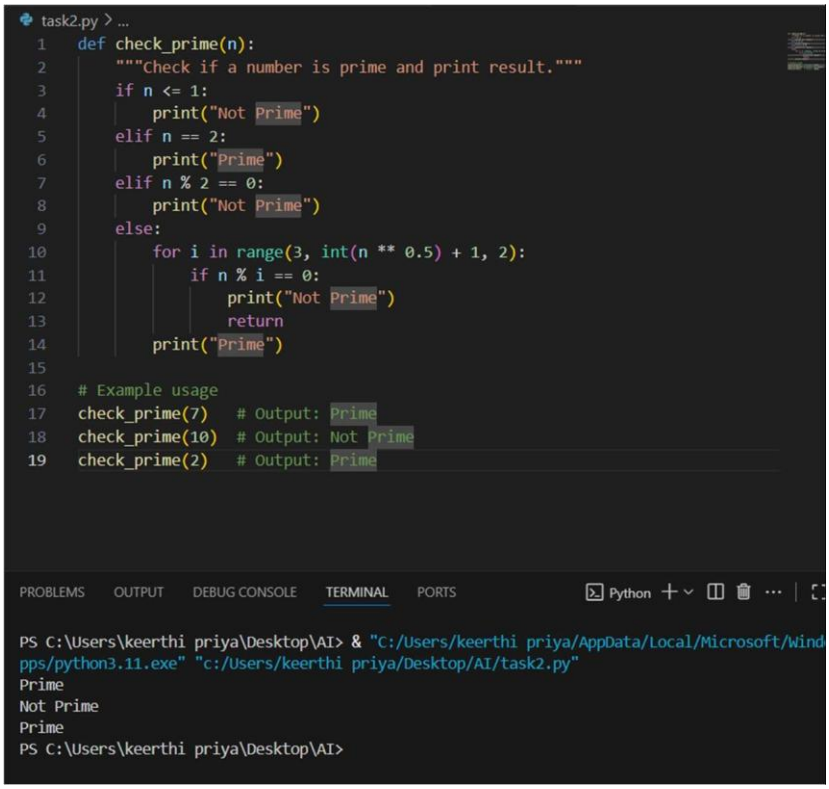
- 

```python
def is_prime(n):
    """Check if a number is prime and print result."""
    if n <= 1:
        print("Not Prime")
        return False
    if n == 2:
        print("Prime")
        return True
    if n % 2 == 0:
        print("Not Prime")
        return False
    for i in range(3, int(n ** 0.5) + 1, 2):
        if n % i == 0:
            print("Not Prime")
            return False
    print("Prime")
    return True

# Example usage
is_prime(5)  # Output: Prime
```

```
PS C:\Users\keerthi priya\Desktop\AI> & "C:/Users/keerthi priya/AppData/Local/Microsoft/WindowsApps/python3.11.exe" "c:/Users/keerthi priya/Desktop/AI/task2.py"
Prime
PS C:\Users\keerthi priya\Desktop\AI>
```

- FEW-SHOT:

- 

```python
def check_prime(n):
    """Check if a number is prime and print result."""
    if n <= 1:
        print("Not Prime")
    elif n == 2:
        print("Prime")
    elif n % 2 == 0:
        print("Not Prime")
    else:
        for i in range(3, int(n ** 0.5) + 1, 2):
            if n % i == 0:
                print("Not Prime")
                return
        print("Prime")

# Example usage
check_prime(7)   # Output: Prime
check_prime(10)  # Output: Not Prime
check_prime(2)   # Output: Prime
```

```
PS C:\Users\keerthi priya\Desktop\AI> & "C:/Users/keerthi priya/AppData/Local/Microsoft/Windows/python3.11.exe" "c:/Users/keerthi priya/Desktop/AI/task2.py"
Prime
Not Prime
Prime
PS C:\Users\keerthi priya\Desktop\AI>
```

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:
    - Data Consumed (in GB) o Plan Type (Prepaid / Postpaid)
    - Additional Services Used (e.g., caller tune, OTT subscription, etc.)
3. Implement billing logic to calculate:
    - DC (Data Charges) – charges based on data consumption
    - VC (Value-added Charges) – charges for additional services
    - Tax – applicable tax on the total bill
4. Display an itemized bill showing:
    - Plan Type o Data Usage and Charges o Value-added Services and Charges o Tax
    - 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.

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.
-

```python
def calculate_mobile_bill():
    print("Mobile Phone Bill Calculator")
    plan_type = input("Enter plan type (Prepaid/Postpaid): ").strip().capit
    data_gb = float(input("Enter data consumed (in GB): "))
    vas_services = input("Enter value-added services used (comma separated,

    # Define rates
    if plan_type == "Prepaid":
        data_rate = 10  # per GB
        vas_rates = {"Caller Tune": 30, "Roaming": 50, "International SMS":
    elif plan_type == "Postpaid":
        data_rate = 8  # per GB
        vas_rates = {"Caller Tune": 25, "Roaming": 40, "International SMS":
    else:
        print("Invalid plan type.")
        return

    # Calculate data charges
    data_charges = data_gb * data_rate

    # Calculate value-added service charges
    vas_list = [service.strip() for service in vas_services.split(",") if s
    vas_charges = 0
```

-

```python
def calculate_mobile_bill():
    vas_details = []
    for service in vas_list:
        charge = vas_rates.get(service, 0)
        vas_charges += charge
        vas_details.append(f"{service}: ₹{charge}")

    # Tax calculation (e.g., 18%)
    subtotal = data_charges + vas_charges
    tax = subtotal * 0.18
    total = subtotal + tax

    # Display bill
    print("\n--- Itemized Mobile Phone Bill ---")
    print(f"Plan Type: {plan_type}")
    print(f"Data Used: {data_gb} GB")
    print(f"Data Charges: ₹{data_charges:.2f}")
    print("Value-Added Services:")
    if vas_details:
        for detail in vas_details:
            print(f"  {detail}")
    else:
        print("  None")
```

task2.py > ...

```
1   def calculate_mobile_bill():
44      else:
45          print("  None")
46      print(f"Value-Added Service Charges: ₹{vas_charges:.2f}")
47      print(f"Tax (18%): ₹{tax:.2f}")
48      print(f"Total Bill Amount: ₹{total:.2f}")
49
50  # Example usage
51  if __name__ == "__main__":
52      calculate_mobile_bill()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS C:\Users\keerthi priya\Desktop\AI> & "C:/Users/keerthi priya/AppData/Local/Microsoft/WindowsA
pps/python3.11.exe" "c:/Users/keerthi priya/Desktop/AI/task2.py"
Mobile Phone Bill Calculator
Enter plan type (Prepaid/Postpaid): 349
Enter data consumed (in GB): 200
Enter value-added services used (comma separated, e.g., Caller Tune,Roaming): sms pack,data
Invalid plan type.
PS C:\Users\keerthi priya\Desktop\AI>
```

o

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)	Week2 - Monday

	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) 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: o AI prompts used to generate the program o AI-generated Python code o Line-by-line explanation of the code	

task2.py > ...

```python
def calculate_lpg_bill():
    print("LPG Cylinder Bill Calculator")
    print("Cylinder Types: Domestic 14.2 kg, Domestic 5 kg, Commercial 19 k
    cylinder_type = input("Enter Cylinder Type: ").strip()
    num_cylinders = int(input("Enter Number of Cylinders Booked: "))

    # Price list
    prices = {
        "Domestic 14.2 kg": 905.00,
        "Domestic 5 kg": 335.50,
        "Commercial 19 kg": 1886.50,
        "Commercial 47.5 kg": 4712.00
    }

    # Get price per cylinder
    price_per_cylinder = prices.get(cylinder_type)
    if price_per_cylinder is None:
        print("Invalid Cylinder Type.")
        return

    # Subsidy (only for domestic)
    subsidy = 0.0
    if cylinder_type.startswith("Domestic"):
```

o o ..

task2.py > ...

```python
def calculate_lpg_bill():
            subsidy = float(input("Enter Subsidy Amount (₹): "))

        # Delivery charges
        delivery_charges = float(input("Enter Delivery Charges (₹10 f

        # Calculate bill
        base_amount = price_per_cylinder * num_cylinders
        total_bill = base_amount - subsidy + delivery_charges

        # Display itemized bill
        print("\n--- Itemized LPG Cylinder 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}")

    # Example usage
    if __name__ == "__main__":
        calculate_lpg_bill()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS C:\Users\keerthi priya\Desktop\AI> & "C:/Users/keerthi priya/AppData/Loca
pps/python3.11.exe" "c:/Users/keerthi priya/Desktop/AI/task2.py"
LPG Cylinder Bill Calculator
Cylinder Types: Domestic 14.2 kg, Domestic 5 kg, Commercial 19 kg, Commercia
Enter Cylinder Type: oblique
Enter Number of Cylinders Booked: 22
Invalid Cylinder Type.
PS C:\Users\keerthi priya\Desktop\AI>
```