

	<h2 style="text-align: center;">AI ASSISTED CODING</h2> <p>NAME: Bayana Arjun          ROLL NO: 2403A510A9          ASSIGNMENT: 3.1</p>	
	<p style="text-align: center;">Lab Experiment: Prompt Engineering – Improving Prompts and Context Management (0.5 marks)</p> <p><b>Objective</b></p> <p>To explore how prompt design and context influence AI-generated outputs and to learn techniques to improve AI responses.</p> <hr/> <p><b>Tools Required</b></p> <ul style="list-style-type: none"> <li>● GitHub Copilot / Google Gemini / ChatGPT</li> <li>● VS Code / Google Colab</li> <li>● Internet access</li> </ul> <p><b>Procedure</b></p> <ol style="list-style-type: none"> <li>1. Select a simple task: "Write a Python function to check if a number is prime."</li> <li>2. Use different prompting strategies to generate the solution:             <ol style="list-style-type: none"> <li>a) Zero-Shot – no examples.</li> <li>b) One-Shot – one example provided.</li> <li>c) Few-Shot – multiple examples provided.</li> <li>d) Context-Managed – detailed prompt with constraints and instructions.</li> </ol> </li> <li>3. Record AI responses and refine prompts to improve code quality.</li> <li>4. Request AI to optimize the logic for efficiency.</li> <li>5. Compare results and document improvements.</li> </ol> <p><b>Sample Prompts ●</b></p> <p>Zero-Shot:</p> <p style="padding-left: 40px;">Write a Python function to check if a number is prime.</p>	

- 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

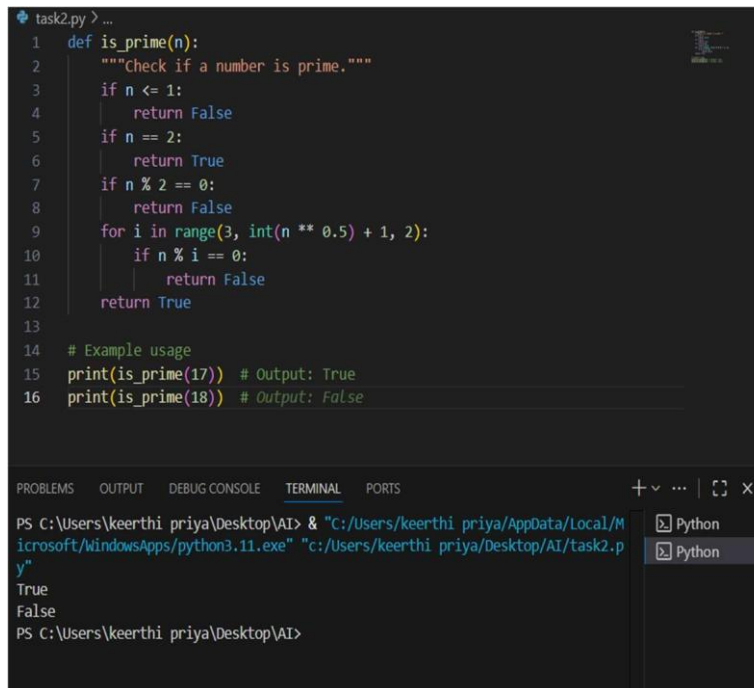
Example 2: Input: 10 → Output: Not Prime

Example 3: Input: 2 → Output: Prime

Generate the function accordingly. ●

Context-Managed (With Optimization)

- ZERO-SHOT:



```
task2.py > ...
1 def is_prime(n):
2     """Check if a number is prime."""
3     if n <= 1:
4         return False
5     if n == 2:
6         return True
7     if n % 2 == 0:
8         return False
9     for i in range(3, int(n ** 0.5) + 1, 2):
10         if n % i == 0:
11             return False
12     return True
13
14 # Example usage
15 print(is_prime(17)) # Output: True
16 print(is_prime(18)) # Output: False
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
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"
True
False
PS C:\Users\keerthi priya\Desktop\AI>
```

- 
- ONE-SHOT:

```
task2.py > ...  
1 def is_prime(n):  
2     """check if a number is prime and print result."""  
3     if n <= 1:  
4         print("Not Prime")  
5         return False  
6     if n == 2:  
7         print("Prime")  
8         return True  
9     if n % 2 == 0:  
10        print("Not Prime")  
11        return False  
12    for i in range(3, int(n ** 0.5) + 1, 2):  
13        if n % i == 0:  
14            print("Not Prime")  
15            return False  
16    print("Prime")  
17    return True  
18  
19 # Example usage  
20 is_prime(5) # Output: Prime
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] [x] [ ] [x]

```
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"  
Prime  
PS C:\Users\keerthi priya\Desktop\AI>
```

## FEW-SHOT:

```
task2.py > ...
1 def check_prime(n):
2     """Check if a number is prime and print result."""
3     if n <= 1:
4         print("Not Prime")
5     elif n == 2:
6         print("Prime")
7     elif n % 2 == 0:
8         print("Not Prime")
9     else:
10        for i in range(3, int(n ** 0.5) + 1, 2):
11            if n % i == 0:
12                print("Not Prime")
13                return
14        print("Prime")
15
16 # Example usage
17 check_prime(7) # Output: Prime
18 check_prime(10) # Output: Not Prime
19 check_prime(2) # Output: Prime
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Python

+

-

□

🗑

⋮

⌂

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



	<p>Task: Mobile Data Usage Billing Application (1.0 Marks)</p> <p>Objective:</p> <p>Use Python programming and AI-assisted coding tools to create an application that simulates mobile data billing for a telecom service provider.</p> <p>Instructions</p> <ol style="list-style-type: none"> <li>1. Use GitHub Copilot or Google Gemini to assist in writing the program.</li> <li>2. Read the following inputs from the user: <ul style="list-style-type: none"> <li>o Data Consumed (in GB)</li> <li>o Plan Type (Prepaid / Postpaid)</li> <li>o Additional Services Used (e.g., caller tune, OTT subscription, etc.)</li> </ul> </li> <li>3. Implement billing logic to calculate: <ul style="list-style-type: none"> <li>o DC (Data Charges) – charges based on data consumption</li> <li>o VC (Value-added Charges) – charges for additional services</li> <li>o Tax – applicable tax on the total bill</li> </ul> </li> <li>4. Display an itemized bill showing: <ul style="list-style-type: none"> <li>o Plan Type</li> <li>o Data Usage and Charges</li> <li>o Value-added Services and Charges</li> <li>o Tax</li> <li>o Total Bill Amount</li> </ul> </li> </ol> <p>Requirements</p> <ul style="list-style-type: none"> <li>• Students must refer to their actual mobile bill for charge structure (data cost, service fees, taxes) to make the program realistic.</li> <li>• AI assistance (Copilot/Gemini) must be used to generate and refine the initial code.</li> </ul> <p>Deliverables</p> <ul style="list-style-type: none"> <li>• AI prompts used for code generation.</li> <li>• AI-generated Python code and any optimized version.</li> <li>• Screenshots of:</li> </ul>	
--	--	--

- AI interactions
- Program execution and output
- Comparison with the student's actual mobile bill.

```
task2.py > ...
1 def calculate_mobile_bill():
2     print("Mobile Phone Bill Calculator")
3     plan_type = input("Enter plan type (Prepaid/Postpaid): ").strip().capitalize()
4     data_gb = float(input("Enter data consumed (in GB): "))
5     vas_services = input("Enter value-added services used (comma separated,")
6
7     # Define rates
8     if plan_type == "Prepaid":
9         data_rate = 10 # per GB
10        vas_rates = {"Caller Tune": 30, "Roaming": 50, "International SMS":
11    elif plan_type == "Postpaid":
12        data_rate = 8 # per GB
13        vas_rates = {"Caller Tune": 25, "Roaming": 40, "International SMS":
14    else:
15        print("Invalid plan type.")
16        return
17
18    # Calculate data charges
19    data_charges = data_gb * data_rate
20
21    # Calculate value-added service charges
22    vas_list = [service.strip() for service in vas_services.split(",") if s
23    vas_charges = 0
```

o

```
task2.py > ...
1 def calculate_mobile_bill():
24     vas_details = []
25     for service in vas_list:
26         charge = vas_rates.get(service, 0)
27         vas_charges += charge
28         vas_details.append(f"{service}: ₹{charge}")
29
30     # Tax calculation (e.g., 18%)
31     subtotal = data_charges + vas_charges
32     tax = subtotal * 0.18
33     total = subtotal + tax
34
35     # Display bill
36     print("\n--- Itemized Mobile Phone Bill ---")
37     print(f"Plan Type: {plan_type}")
38     print(f>Data Used: {data_gb} GB")
39     print(f>Data Charges: ₹{data_charges:.2f}")
40     print("Value-Added Services:")
41     if vas_details:
42         for detail in vas_details:
43             print(f"    {detail}")
44     else:
45         print("    None")
```

o



3	<p>Task: Develop an LPG Billing System (1.0 Marks)</p> <p>Objective</p> <p>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.</p> <p>Instructions</p> <ol style="list-style-type: none"> <li>1. Use GitHub Copilot or Google Gemini to assist in writing and refining the program.</li> <li>2. Read the following user inputs: <ul style="list-style-type: none"> <li>o Cylinder Type (Domestic 14.2 kg / Domestic 5 kg / Commercial 19 kg / Commercial 47.5 kg) o</li> <li>Number of Cylinders Booked o</li> <li>Subsidy Amount (applicable only for domestic cylinders)</li> </ul> </li> </ol>	Week2 - Monday
---	---	----------------



	<p>3. Refer to the given LPG Price List to determine the price per cylinder:</p> <ul style="list-style-type: none"> <li>o Domestic LPG (14.2 kg) → ₹905.00</li> <li>o Domestic LPG (5 kg) → ₹335.50</li> <li>o Commercial LPG (19 kg) → ₹1,886.50</li> <li>o Commercial LPG (47.5 kg) → ₹4,712.00</li> <li>o Delivery Charges (₹10 to ₹50)</li> </ul> <p>4. Implement the billing formula:</p> <p>Bill Amount = (Price per Cylinder × Quantity) - Subsidy (if applicable) + Delivery Charges</p> <p>5. Calculate and display an itemized bill including:</p> <ul style="list-style-type: none"> <li>• Cylinder Type</li> <li>• Number of Cylinders</li> <li>• Base Amount</li> <li>• Subsidy</li> <li>• Delivery Charges</li> <li>• Total Bill Amount</li> </ul> <p>Deliverables</p> <ul style="list-style-type: none"> <li>• A report containing: <ul style="list-style-type: none"> <li>o AI prompts used to generate the program</li> <li>o AI-generated Python code</li> <li>o Line-by-line explanation of the code</li> </ul> </li> </ul>	
--	--	--

```

task2.py > ...
1 def calculate_lpg_bill():
2     print("LPG Cylinder Bill Calculator")
3     print("Cylinder Types: Domestic 14.2 kg, Domestic 5 kg, Commercial 19 k
4     cylinder_type = input("Enter Cylinder Type: ").strip()
5     num_cylinders = int(input("Enter Number of Cylinders Booked: "))
6
7     # Price list
8     prices = {
9         "Domestic 14.2 kg": 905.00,
10        "Domestic 5 kg": 335.50,
11        "Commercial 19 kg": 1886.50,
12        "Commercial 47.5 kg": 4712.00
13    }
14
15    # Get price per cylinder
16    price_per_cylinder = prices.get(cylinder_type)
17    if price_per_cylinder is None:
18        print("Invalid Cylinder Type.")
19        return
20
21    # Subsidy (only for domestic)
22    subsidy = 0.0
23    if cylinder_type.startswith("Domestic"):

```

```

O O task2.py > ...
1 def calculate_lpg_bill():
24     subsidy = float(input("Enter Subsidy Amount (₹): "))
25
26     # Delivery charges
27     delivery_charges = float(input("Enter Delivery Charges (₹10
28
29     # Calculate bill
30     base_amount = price_per_cylinder * num_cylinders
31     total_bill = base_amount - subsidy + delivery_charges
32
33     # Display itemized bill
34     print("\n--- Itemized LPG Cylinder Bill ---")
35     print(f"Cylinder Type: {cylinder_type}")
36     print(f"Number of Cylinders: {num_cylinders}")
37     print(f"Base Amount: ₹{base_amount:.2f}")
38     print(f"Subsidy: ₹{subsidy:.2f}")
39     print(f"Delivery Charges: ₹{delivery_charges:.2f}")
40     print(f"Total Bill Amount: ₹{total_bill:.2f}")
41
42     # Example usage
43     if __name__ == "__main__":
44         calculate_lpg_bill()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python +

```

PS C:\Users\keerthi priya\Desktop\AI> & "C:/Users/keerthi priya/AppData/Local/Programs/Python/Python311/python.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, Commercial 47.5 kg
Enter Cylinder Type: oblique
Enter Number of Cylinders Booked: 22
Invalid Cylinder Type.
PS C:\Users\keerthi priya\Desktop\AI>

```