

NAME:SIRIVELLA SANJANA

ID:2403A510D4

BATCH:05

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 STUDENT DETAILS:		Venkataramana Veeramsetty. Sirivella sanjana Batch-05 2403a510d4	
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 (Dharmateja) Intern 2 (Sai Prasad) Intern 3 (Sowmya) NS_2 (Mounika)	
Course Code	24CS002PC215	Course Title	AI Assisted Coding
Year/Sem	II/I	Regulation	R24
Date and Day of Assignment	Week 2- Monday	Time(s)	
Duration	2 Hours	Applicable to Batches	24CSBTB01 To 24CSBTB39
Assignment Number: 2.1 (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	Week 2- 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.

1. 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:

Prompt:

 Write a Python function to check if a number is prime

Ask or edit in context

Accept

Close



Code:

```
def is_prime(n):
    if n <= 1:
        return False
    if n == 2:
        return True
    if n % 2 == 0:
        return False
    for i in range(3, int(n ** 0.5) + 1, 2):
        if n % i == 0:
            return False
    return True
```

One-shot:

Prompt:



write a function to check if a number is prime with one example

Ask or edit in context

Accept

Close



Code:

```
def is_prime(n):
    if n <= 1:
        return
    for i in range(2, n):
        if n % i == 0:
            return
    print("Prime")


# Example usage
number = int(input("Enter a number:"))
is_prime(number)
```

Output:


```
C:\Users\venub\OneDrive\Desktop\AIAC_Lab\Lab_3>python 3
Enter a number:5
Prime
```

Few-shot:

Prompt:

 Write a Python function to check if a number is prime

Ask or edit in context

Accept Close  

Code:

```
def check_prime(n):
    if n <= 1:
        return "Not Prime"
    if n == 2:
        return "Prime"
    if n % 2 == 0:
        return "Not Prime"
    for i in range(3, int(n**0.5) + 1):
        if n % i == 0:
            return "Not Prime"
    return "Prime"

num = int(input("Enter a number: "))
print(check_prime(num))
```

Output:


```
C:\Users\venub\OneDrive\Desktop\AIAC_Lab\Lab_3>python 3_1
Enter a number: 6
Not Prime

C:\Users\venub\OneDrive\Desktop\AIAC_Lab\Lab_3>python 3_1
Enter a number: 7
Prime



C:\Users\venub\OneDrive\Desktop\AIAC_Lab\Lab_3>python 3_1
Enter a number: 5
Prime
```

Context-Managed:

Prompt:

 Write a Python function to check if a number is prime

Ask or edit in context

Accept Close  

Code:

```
def check_prime(n):
    if n < 0:
        return "Not Prime"
    if n in (0, 1):
        return "Neither Prime Nor Composite"
    if n <= 3:
        return "Prime"
    if n % 2 == 0 or n % 3 == 0:
        return "Not Prime"
    i = 5
    while i * i <= n:
        if n % i == 0 or n % (i + 2) == 0:
            return "Not Prime"
        i += 6
    return "Prime"

# Example usage with output
number = int(input("Enter a number: "))
print(check_prime(number))
```

Output:

```
C:\Users\venub\OneDrive\Desktop\AIAC_Lab\Lab_3>python 3_1.  
Enter a number: -1  
Not Prime  
  
C:\Users\venub\OneDrive\Desktop\AIAC_Lab\Lab_3>python 3_1.  
Enter a number: 0  
Neither Prime Nor Composite  
  
C:\Users\venub\OneDrive\Desktop\AIAC_Lab\Lab_3>python 3_1.  
Enter a number: 2  
Prime
```

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)
 - 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
 - Data Usage and Charges
 - Value-added Services and Charges
 - 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

Week2 -
Monday

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.

prompt:



Write a Python program that calculates a user's mobile bill. The program should take as input the user's mobile data consumed (in GB), plan type (Prepaid/Postpaid), and additional services (like caller tune, ott). The program should compute data charges, value-added charges, tax, and total. Output a detailed bill.

Ask or edit in context

Cancel



Code:

```
# Mobile Bill Calculator
```

```
def get_float(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
        print("Invalid plan type. Please enter 'Prepaid' or 'Postpaid'.")

def get_services():
    services = []
    print("Select additional services (type 'done' or press Enter when done).")
    print("Options: caller tune, ott")
    while True:
        service = input("Add service: ").strip().lower()
        if service == "done" or service == "":
            break
        if service in ["caller tune", "ott"]:
            if service not in services:
                services.append(service)
```

```
        services.append(service)
    else:
        print("Service already added.")
    else:
        print("Invalid service.")
return services

def calculate_data_charges(data_gb, plan_type):
    # Example rates
    if plan_type == "prepaid":
        rate_per_gb = 10.0
    else:
        rate_per_gb = 8.0
    return data_gb * rate_per_gb

def calculate_value_added_charges(services):
    charges = 0.0
    service_rates = {"caller tune": 2.0, "ott": 5.0}
    for service in services:
        charges += service_rates.get(service, 0.0)
    return charges

def calculate_tax(amount):
    tax_rate = 0.18 # 18% GST
    return amount * tax_rate
```



```

def print_bill(data_gb, plan_type, services, data_charges, value_a
    print("\n--- Mobile Bill ---")
    print(f"Plan Type: {plan_type.capitalize()}")
    print(f>Data Consumed: {data_gb:.2f} GB")
    print(f>Data Charges: ₹{data_charges:.2f}")
    print("Value Added Services:")
    if services:
        for s in services:
            print(f" - {s.capitalize()}")
            print(f"Value Added Charges: ₹{value_added:.2f}")
    else:
        print(" None")
        print(f"Value Added Charges: ₹0.00")
    print(f"Tax (18% GST): ₹{tax:.2f}")
    print(f"Total Amount: ₹{total:.2f}")
    print("-----")

def main():
    print("Welcome to Mobile Bill Calculator")
    data_gb = get_float("Enter data consumed (in GB): ")
    plan_type = get_plan_type()
    services = get_services()
    data_charges = calculate_data_charges(data_gb, plan_type)
    value_added = calculate_value_added_charges(services)
    subtotal = data_charges + value_added
    tax = calculate_tax(subtotal)
    total = subtotal + tax
    print_bill(data_gb, plan_type, services, data_charges, value_a

```

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:
 - Cylinder Type (Domestic 14.2 kg / Domestic 5 kg / Commercial 19 kg / Commercial 47.5 kg)
 - Number of Cylinders Booked

- Subsidy Amount (applicable only for domestic cylinders)
3. Refer to the given LPG Price List to determine the price per cylinder:
- Domestic LPG (14.2 kg) → ₹905.00
 - Domestic LPG (5 kg) → ₹335.50
 - Commercial LPG (19 kg) → ₹1,886.50
 - Commercial LPG (47.5 kg) → ₹4,712.00

Week2 -

Monday

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

prompt:

```
Write a Python program to calculate LPG bills. It should ask for cylinder type (Domestic 14.2kg, Domestic 5kg, Commercial 19kg, Commercial 47.5kg), quantity, subsidy amount (only for domestic cylinders), and delivery charges (between ₹10 and ₹50). Calculate the bill as (Price per Cylinder × Quantity) - Subsidy + delivery_charges. Output an itemized bill with cylinder type, quantity, base amount, subsidy, delivery charges, and total bill amount.
```

Ask or edit in context

Cancel

Code:

```
# LPG Bill Calculator

def get_cylinder_info():
    cylinder_types = {
        "1": ("Domestic 14.2kg", 1050),
        "2": ("Domestic 5kg", 400),
        "3": ("Commercial 19kg", 1800),
        "4": ("Commercial 47.5kg", 4500)
    }
    print("Select Cylinder Type:")
    for key, (name, price) in cylinder_types.items():
        print(f"{key}. {name} (₹{price})")
    choice = input("Enter choice (1-4): ").strip()
    if choice not in cylinder_types:
        print("Invalid choice.")
        exit(1)
    return cylinder_types[choice]

def get_int_input(prompt, min_val=1):
    try:
        val = int(input(prompt))
        if val < min_val:
            raise ValueError
        return val
    except ValueError:
        print("Invalid input.")
        exit(1)
```

```

def get_float_input(prompt, min_val=0, max_val=None):
    try:
        val = float(input(prompt))
        if val < min_val or (max_val is not None and val > max_val):
            raise ValueError
        return val
    except ValueError:
        print("Invalid input.")
        exit(1)

def main():
    cylinder_type, price_per_cylinder = get_cylinder_info()
    quantity = get_int_input("Enter number of cylinders: ", 1)
    subsidy = 0
    if "Domestic" in cylinder_type:
        subsidy = get_float_input("Enter subsidy amount (₹): ", 0)
    delivery_charges = get_float_input("Enter delivery charges (₹10-₹50): ")
    base_amount = price_per_cylinder * quantity
    total = base_amount - subsidy + delivery_charges

    print("\n--- LPG Bill ---")
    print(f"Cylinder Type      : {cylinder_type}")
    print(f"Quantity              : {quantity}")
    print(f"Base Amount           : ₹{base_amount:.2f}")
    print(f"Subsidy               : ₹{subsidy:.2f}")
    print(f"Delivery Charges      : ₹{delivery_charges:.2f}")
    print(f"Total Bill            : ₹{total:.2f}")

```

Output:

```

C:\Users\venub\OneDrive\Desktop\AIAC_Lab\Lab_3>python 3_1
Select Cylinder Type:
1. Domestic 14.2kg (₹1050)
2. Domestic 5kg (₹400)
3. Commercial 19kg (₹1800)
4. Commercial 47.5kg (₹4500)
Enter choice (1-4): 2
Enter number of cylinders: 7
Enter subsidy amount (₹): 200
Enter delivery charges (₹10-₹50): 25

--- LPG Bill ---
Cylinder Type      : Domestic 5kg
Quantity           : 7
Base Amount        : ₹2800.00
Subsidy            : ₹200.00
Delivery Charges   : ₹25.00
Total Bill         : ₹2625.00

```

--	--	--

