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)**	Week2 - Monday

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:
- Prompt:
  Write a Python function to check if a number is prime.

Code:

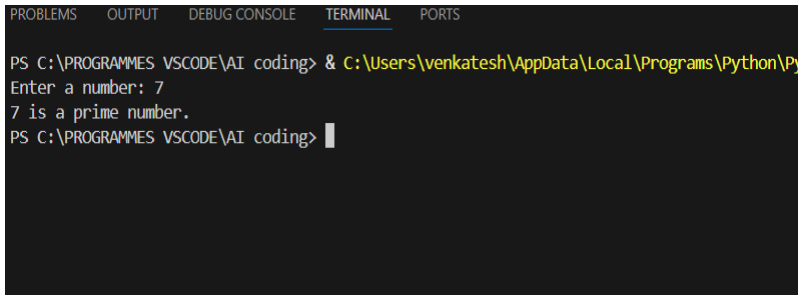
```python
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

# Get user input
num = int(input("Enter a number: "))

# Check if the number is prime
if is_prime(num):
    print(f"{num} is a prime number.")
else:
    print(f"{num} is not a prime number.")
```

t:

Output:

- 

- One-Shot:

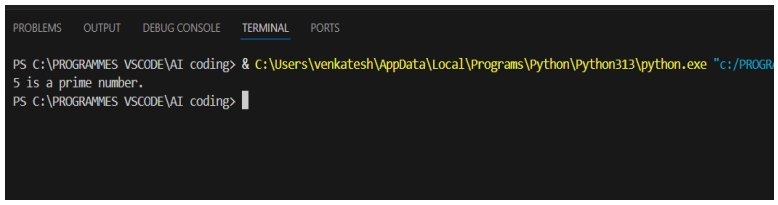
  Example: Input: 5 → Output: Prime. Now, write a function to check if a number is prime.

- Prompt:
- Write a py function to check 5 is a prime or not.
- Code:

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

# Check if 5 is prime
if is_prime(5):
    print("5 is a prime number.")
else:
    print("5 is not a prime number.")
```

- Output:



- 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)
- Prompt;
- Write a py function to check 7,10,2 ia prime or not.
- Code:

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

# Check if 7, 10, and 2 are prime
numbers = [7, 10, 2]
for num in numbers:
    if is_prime(num):
        print(f"{num} is a prime number.")
    else:
        print(f"{num} is not a prime number.")
```

- Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\PROGRAMMES VSCODE\AI coding> & C:\Users\venkatesh\AppData\Local\Programs\Python\Python
7 is a prime number.
10 is not a prime number.
2 is a prime number.
PS C:\PROGRAMMES VSCODE\AI coding>
```

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:
	  ○ Data Consumed (in GB)
	  ○ Plan Type (Prepaid / Postpaid)
	  ○ Additional Services Used (e.g., caller tune, OTT subscription, etc.)
	3. Implement billing logic to calculate:
		Week2 - Monday

- DC (Data Charges) – charges based on data consumption
- VC (Value-added Charges) – charges for additional services
- Tax – applicable tax on the total bill

Code:

-
```python
# Telecom Billing System

# Step 1: Read inputs
data_consumed = float(input("Enter Data Consumed (in GB): "))
plan_type = input("Enter Plan Type (Prepaid/Postpaid): ").strip().lower()
services = input("Enter Additional Services Used (comma-separated, e.g., caller tune, ott): ").strip().lower(

# Step 2: Define charges
# Data charges (example values)
if plan_type == "prepaid":
    data_rate = 10   # Rs. per GB
elif plan_type == "postpaid":
    data_rate = 8    # Rs. per GB (cheaper than prepaid)
else:
    print("Invalid plan type! Defaulting to prepaid rates.")
    data_rate = 10
```
-
```python
# Additional service charges
service_charges = {
    "caller tune": 30,
    "ott": 100,
    "roaming": 50
}

# Step 3: Calculate charges
# Data Charges
DC = data_consumed * data_rate

# Value-added Charges
VC = 0
for s in services:
    s = s.strip()
    if s in service_charges:
        VC += service_charges[s]

# Tax calculation (let's assume 18%)
subtotal = DC + VC
tax = subtotal * 0.18

# Final Bill
total_bill = subtotal + tax

```
-
```python
tax = subtotal * 0.18

# Final Bill
total_bill = subtotal + tax

# Step 4: Print detailed bill
print("\n--- Telecom Bill ---")
print(f"Data Consumed: {data_consumed} GB")
print(f"Plan Type: {plan_type.capitalize()}")
print(f"Data Charges (DC): ₹{DC:.2f}")
print(f"Value-added Charges (VC): ₹{VC:.2f}")
print(f"Tax (18%): ₹{tax:.2f}")
print(f"Total Bill: ₹{total_bill:.2f}")

```
- Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Total Bill: ₹23.60
PS C:\PROGRAMMES VSCODE\AI coding> & C:\Users\venkatesh\AppData\Local\Programs\Python\Python313\python.exe "c:/
Enter Data Consumed (in GB): 2
Enter Plan Type (Prepaid/Postpaid): prepaid
Enter Additional Services Used (comma-separated, e.g., caller tune, ott): caller

--- Telecom Bill ---
Data Consumed: 2.0 GB
Plan Type: Prepaid
Data Charges (DC): ₹20.00
Value-added Charges (VC): ₹0.00
Tax (18%): ₹3.60
Total Bill: ₹23.60
```

- 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 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;

Create py code of display items bill plan type and data useses and value added and tax and total bill amount.

	Code:
	`PS C:\PROGRAMMES VSCODE\AI coding> & C:\Users\venkatesh\AppData\Local\Programs\Python\Python313\python.exe "c:/PROGR`
	`PS C:\PROGRAMMES VSCODE\AI coding> & C:\Users\venkatesh\AppData\Local\Programs\Python\Python313\python.exe "c:/PROGR`
	`PS C:\PROGRAMMES VSCODE\AI coding> & C:\Users\venkatesh\AppData\Local\Programs\Python\Python313\python.exe "c:/PROGR`
	`Enter Data Used (GB): 2`
	`Enter Plan Type (Prepaid/Postpaid): postpaid`
	`Enter Services (comma-separated, e.g., caller tune, ott): ott`
	
	`--- ITEMIZED BILL ---`
	`Plan Type: Postpaid`
	`Data: 2.0 GB → ₹16.00`
	`Value-added Services: ₹149.00`
	`Tax (18%): ₹29.70`
	`Total Bill: ₹194.70`
	`PS C:\PROGRAMMES VSCODE\AI coding>`
	e
	Output:
	`PS C:\PROGRAMMES VSCODE\AI coding> & C:\Users\venkatesh\AppData\Local\Programs\Python\Python313\python.exe "c:/PROGR`
	`PS C:\PROGRAMMES VSCODE\AI coding> & C:\Users\venkatesh\AppData\Local\Programs\Python\Python313\python.exe "c:/PROGR`
	`PS C:\PROGRAMMES VSCODE\AI coding> & C:\Users\venkatesh\AppData\Local\Programs\Python\Python313\python.exe "c:/PROGR`
	`Enter Data Used (GB): 2`
	`Enter Plan Type (Prepaid/Postpaid): postpaid`
	`Enter Services (comma-separated, e.g., caller tune, ott): ott`
	
	`--- ITEMIZED BILL ---`
	`Plan Type: Postpaid`
	`Data: 2.0 GB → ₹16.00`
	`Value-added Services: ₹149.00`
	`Tax (18%): ₹29.70`
	`Total Bill: ₹194.70`
	`PS C:\PROGRAMMES VSCODE\AI coding>`
	●	

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	Week2 - Monday

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
    - 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: generate a py code of LPG cylinder bill system.

Code:

```python
prices = {
    "domestic 14.2": 905.00,
    "domestic 5": 335.50,
    "commercial 19": 1886.50,
    "commercial 47.5": 4712.00
}

# User input
cylinder = input("Enter Cylinder Type (Domestic 14.2 / Domestic 5 / Commercial 19 / Commercial 47.5): ").low
qty = int(input("Enter Number of Cylinders: "))
subsidy = float(input("Enter Subsidy Amount (0 if not applicable): "))
delivery = float(input("Enter Delivery Charges (10-50): "))

# Bill calculation
base = prices.get(cylinder, 0) * qty
bill = base - subsidy + delivery

# Itemized bill
print("\n--- LPG BILL ---")
print(f"Cylinder Type   : {cylinder.title()}")
print(f"No. of Cylinders: {qty}")
print(f"Base Amount     : ₹{base:.2f}")
print(f"Subsidy         : ₹{subsidy:.2f}")
print(f"Delivery Charges: ₹{delivery:.2f}")
print(f"Total Bill      : ₹{bill:.2f}")
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\PROGRAMMES VSCODE\AI coding> & C:\Users\venkatesh\AppData\Local\Programs\Python\Python313\python.exe "c:/PROG
Enter Cylinder Type (Domestic 14.2 / Domestic 5 / Commercial 19 / Commercial 47.5): Domestic 5
Enter Number of Cylinders: 5
Enter Subsidy Amount (0 if not applicable): 500
Enter Delivery Charges (10-50): 50

--- LPG BILL ---
Cylinder Type   : Domestic 5
No. of Cylinders: 5
Base Amount     : ₹1677.50
Subsidy         : ₹500.00
Delivery Charges: ₹50.00
Total Bill      : ₹1227.50
PS C:\PROGRAMMES VSCODE\AI coding>
```

Comment:

LPG Billing System

This program calculates the LPG bill based on:

- Cylinder type and its price
- Number of cylinders booked
- Subsidy amount (if applicable)
- Delivery charges

It then prints an itemized bill showing all details and the total payable amount.