

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		Dr. Rishabh Mittal	
Instructor(s) Name		Mr. S Naresh Kumar	
		Ms. B. Swathi	
		Dr. Sasanko Shekhar Gantayat	
		Mr. Md Sallauddin	
		Dr. Mathivanan	
		Mr. Y Srikanth	
		Ms. N Shilpa	
		Dr. Rishabh Mittal (Coordinator)	
		Dr. R. Prashant Kumar	
		Mr. Ankushavali MD	
		Mr. B Viswanath	
		Ms. Sujitha Reddy	
		Ms. A. Anitha	
		Ms. M.Madhuri	
		Ms. Katherashala Swetha	
		Ms. Velpula sumalatha	
		Mr. Bingi Raju	
Course Code	23CS002PC304	Course Title	AI Assisted Coding
Year/Sem	III/I	Regulation	R23
Date and Day of Assignment	Week 2 - Wednesday	Time(s)	23CSBTB01 To 23CSBTB52
Duration	2 Hours	Applicable to Batches	All batches
Assignment Number: 3.3(Present assignment number)/24(Total number of assignments)			
Q . N o .	Question		Expected Time to complete
1	Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques Lab Objectives <ul style="list-style-type: none"> To explore and apply different levels of prompt examples in AI-assisted code generation To understand how zero-shot, one-shot, and few-shot prompting affect AI output quality To evaluate the impact of context richness and example quantity on AI performance To build awareness of prompt strategy effectiveness for different problem types 		Week2 - Wednesday

Lab Outcomes (LOs)

After completing this lab, students will be able to:

- Use zero-shot prompting to instruct AI with minimal context
- Use one-shot prompting with a single example to guide AI code generation
- Apply few-shot prompting using multiple examples to improve AI responses
- Compare AI outputs across different prompting strategies

Task 1: Zero-Shot Prompting – Leap Year Check

Scenario

Zero-shot prompting involves giving instructions without providing examples.

Task Description

Use zero-shot prompting to instruct an AI tool to generate a Python function that:

- Accepts a year as input
- Checks whether the given year is a leap year
- Returns an appropriate result

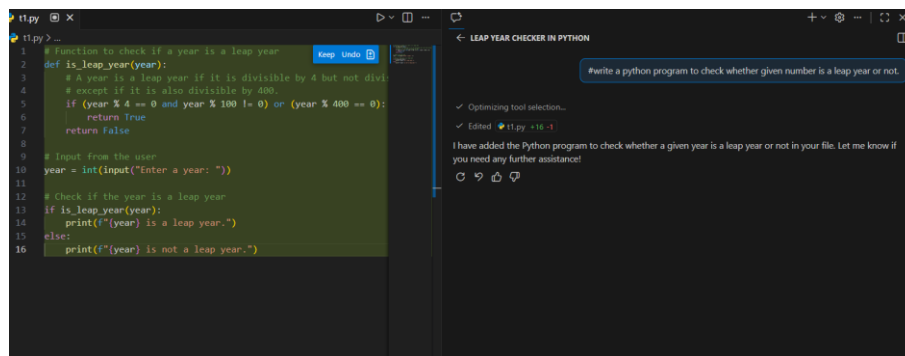
Note: No input-output examples should be provided in the prompt.

Expected Output

- AI-generated leap year checking function
- Correct logical conditions
- Sample input and output
- Screenshot of AI-generated response (if required)

Prompt:

#write a python program to check whether given number is a leap year or not.



```
PS C:\Users\PC\Documents\web projects\CC\AI ASST\AI 3.4\New folder> & C:/Python3
12/python.exe "c:/Users/PC/Documents/web projects/CC/AI ASST/AI 3.4/New folder/t
1.py"
Enter a year: 2026
2026 is not a leap year.
PS C:\Users\PC\Documents\web projects\CC\AI ASST\AI 3.4\New folder> & C:/Python3
12/python.exe "c:/Users/PC/Documents/web projects/CC/AI ASST/AI 3.4/New folder/t
1.py"
Enter a year: 2020
2020 is a leap year.
PS C:\Users\PC\Documents\web projects\CC\AI ASST\AI 3.4\New folder>
```

Observation

Even without examples, the AI correctly:

- Applied leap year rules
- Used logical operators properly
- Returned accurate results

This shows that **zero-shot prompting works well for well-known logical problems.**

Task 2: One-Shot Prompting – Centimeters to Inches Conversion

Scenario

One-shot prompting guides AI using a single example.

Task Description

Use one-shot prompting by providing one input-output example to generate a Python function that:

- Converts centimeters to inches
- Uses the correct mathematical formula

Example provided in prompt:

Input: 10 cm → Output: 3.94 inches

Expected Output

- Python function with correct conversion logic
- Accurate calculation

- Sample test cases and outputs

Prompt

Write a py program to convert cm to inches.

-

```
t1.py
1 # Write a Python function to convert centimeters to inches.
2 # Example:
3 # Input: 10 cm
4 # Output: 3.94 inches
5 def cm_to_inches(cm):
6     inches = cm / 2.54
7     return round(inches, 2)
8
9
10 print(cm_to_inches(25)) # Output: 9.84 inches
```

```
PS C:\Users\PC\Documents\web projects\CC\AI ASST\AI 3.4\New folder> & C:/Python312/python.exe "c:/Users/PC/Documents/web projects/CC/AI ASST/AI 3.4/New folder/t1.py"
Enter a year: 2026
2026 is not a leap year.
PS C:\Users\PC\Documents\web projects\CC\AI ASST\AI 3.4\New folder> & C:/Python312/python.exe "c:/Users/PC/Documents/web projects/CC/AI ASST/AI 3.4/New folder/t1.py"
Enter a year: 2020
2020 is a leap year.
PS C:\Users\PC\Documents\web projects\CC\AI ASST\AI 3.4\New folder> & C:/Python312/python.exe "c:/Users/PC/Documents/web projects/CC/AI ASST/AI 3.4/New folder/t1.py"
3.94
PS C:\Users\PC\Documents\web projects\CC\AI ASST\AI 3.4\New folder>
```

Task 3: Few-Shot Prompting – Name Formatting

Scenario

Few-shot prompting improves accuracy by providing multiple examples.

Task Description

Use few-shot prompting with 2–3 examples to generate a Python function that:

- Accepts a full name as input
- Formats it as “Last, First”

Example formats:

- "John Smith" → "Smith, John"
- "Anita Rao" → "Rao, Anita"

Expected Output

- Well-structured Python function
- Output strictly following example patterns
- Correct handling of names
- Sample inputs and outputs
- Prompt

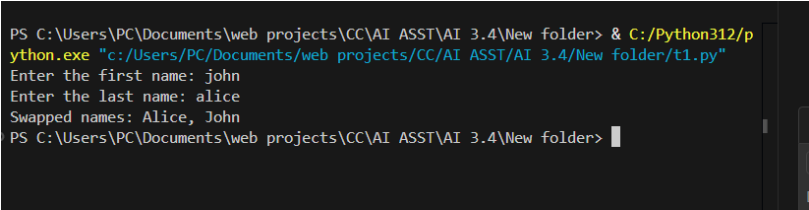
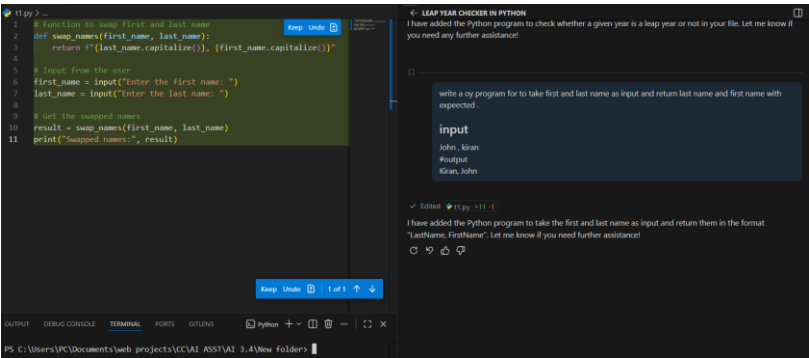
write a py program for to take first and last name as input and return last name and first name with expected .

input

John , kiran

#output

Kiran, John



Task 4: Comparative Analysis – Zero-Shot vs Few-Shot

Scenario

Different prompt strategies may produce different code quality.

Task Description

- Use zero-shot prompting to generate a function that counts vowels in a string
- Use few-shot prompting for the same problem

- Compare both outputs based on:
 - Accuracy
 - Readability
 - Logical clarity

Expected Output

- Two vowel-counting functions
- Comparison table or short reflection paragraph
- Conclusion on prompt effectiveness

Prompt

Write a Python function to count vowels in a string.

Examples:

"hello" → 2

"Python" → 1

The screenshot shows a VS Code editor with a Python script. The script defines two functions: `swap_names` and `count_vowels`. It then takes user input for first and last names, swaps them, and asks for a string to count vowels in. A chat window on the right shows the prompt 'Write a Python function to count vowels in a string.' with examples: 'hello' → 2 and 'Python' → 1.

```
PS C:\Users\PC\Documents\web projects\CC\AI ASST\AI 3.4\New folder> & C:/Python312/python.exe "c:/Users/PC/Documents/web projects/CC/AI ASST/AI 3.4/New folder/t1.py"
Enter the first name: annaakkk
Enter the last name: jsjwi
Swapped names: Jsjwi, Annaakkk
Enter a string: jdqndnqxq
The number of vowels in the string is: 0
PS C:\Users\PC\Documents\web projects\CC\AI ASST\AI 3.4\New folder>
```

Task 5: Few-Shot Prompting – File Handling

Scenario

File processing requires clear logical understanding.

Task Description

Use few-shot prompting to generate a Python function that:

- Reads a .txt file
- Counts the number of lines in the file
- Returns the line count

Expected Output

- Working Python file-processing function
- Correct line count
- Sample .txt input and output
- AI-assisted logic explanation

Note: Report should be submitted as a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots.

The screenshot shows a VS Code editor with a Python file named `tl.py`. The code defines a function `count_lines_in_file` that takes a file path and returns the number of lines. It also includes a main block that prompts the user for a file path and prints the line count. The terminal output shows the script being executed, with the user entering a file path and the script outputting the line count.

```

16 def count_lines_in_file(file_path):
17     return len(lines)
18 except FileNotFoundError:
19     return "File not found. Please check the file path."
20
21 # Input from the user
22 first_name = input("Enter the first name: ")
23 last_name = input("Enter the last name: ")
24
25 # Get the swapped names
26 result = swap_names(first_name, last_name)
27 print("Swapped names:", result)
28
29 # Example usage
30 example_string = input("Enter a string: ")
31 vowel_count = count_vowels(example_string)
32 print(f"The number of vowels in the string is: {vowel_count}")
33
34 # Example usage
35 file_path = input("Enter the file path: ")
36 line_count = count_lines_in_file(file_path)
37 print(f"The number of lines in the file is: {line_count}")

```

The terminal output shows the script being executed, with the user entering a file path and the script outputting the line count.

```

PS C:\Users\PC\Documents\web_projects\CC\AI ASST\AI 3.4\New folder> & C:\Python312\p
python.exe "c:/Users/PC/Documents/web_projects/CC/AI ASST/AI 3.4/New folder/tl.py"
Enter the first name: annaakkk
Enter the last name: jsjui
Swapped names: jsjui, Annaakkk
Enter a string: jdanndhuxq
The number of vowels in the string is: 0
PS C:\Users\PC\Documents\web_projects\CC\AI ASST\AI 3.4\New folder>

```

The right sidebar shows a chat window with the prompt: "Write a Python function to count the number of lines in a text file. Example: File content: Hello World Output: 2". The chat response shows the function definition and the output of the script.