

2303A51225

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		Assignment Type: Lab	
AcademicYear: 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/II	Regulation	R23
Date and Day of Assignment	Week3-Tuesday	Time(s)	
Duration	2 Hours	Applicable to Batches	23CSBTB01 To 23CSBTB52
AssignmentNumber:3.2 (Present assignment number)/ 24 (Total number of assignments)			
Q.No.	Question		Expected Time to complete

Lab 3: Prompt Engineering – Improving Prompts and Context Management

Lab Objectives:

- To understand how prompt structure and wording influence AI-generated code.
- To explore how context (like comments and function names) helps AI generate relevant output.
- To evaluate the quality and accuracy of code based on prompt clarity.
- To develop effective prompting strategies for AI-assisted programming.

Lab Outcomes (LOs):

After completing this lab, students will be able to:

- Generate Python code using Google Gemini in Google Colab.
- Analyze the effectiveness of code explanations and suggestions by Gemini.
- Set up and use Cursor AI for AI-powered coding assistance.
- Evaluate and refactor code using Cursor AI features.
- Compare AI tool behavior and code quality across different platforms.

Task Description-1

- Progressive Prompting for Calculator Design: Ask the AI to design a simple calculator program by initially providing only the function name. Gradually enhance the prompt by adding comments and usage examples.

Expected Output-1

- Comparison showing improvement in AI-generated calculator logic and structure.

Prompt

*"Write a Python calculator program that takes two numbers and an operator (+, -, *, /) from the user and prints the result. Handle division by zero."*

Code

```
[4]
✓ 18s def calculator():
    num1 = float(input("Enter first number: "))
    num2 = float(input("Enter second number: "))
    operator = input("Enter operator (+, -, *, /): ")

    if operator == '+':
        print("Result:", num1 + num2)
    elif operator == '-':
        print("Result:", num1 - num2)
    elif operator == '*':
        print("Result:", num1 * num2)
    elif operator == '/':
        if num2 != 0:
            print("Result:", num1 / num2)
        else:
            print("Error: Division by zero")
    else:
        print("Invalid operator")

calculator()
```

Code output

```
... Enter first number: 12
Enter second number: 5
Enter operator (+, -, *, /): +
Result: 17.0
```

Explanation

- ☐ Progressive prompts improve logic and structure.
- ☐ Clear instructions help AI handle edge cases like division by zero.
- ☐ Code becomes more reliable with better prompts.

