

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
ProgramName: B. Tech		Assignment Type: Lab	AcademicYear: 2025-2026
CourseCoordinatorName		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)	
NS_2 (Mounika)			
CourseCode	24CS002PC215	CourseTitle	AI Assisted Coding
Year/Sem	II/I	Regulation	R24
Date and Day of Assignment	Week1 - Tuesday	Time(s)	
Duration	2 Hours	Applicable to Batches	24CSBTB01 To 24CSBTB39
AssignmentNumber: 1.2 (Present assignment number) / 24 (Total number of assignments)			
Q.No.	Question	Expected Time to complete	
1	Lab 1: Environment Setup – GitHub Copilot and VS Code Integration Lab Objectives: <ul style="list-style-type: none"> To install and configure GitHub Copilot in Visual Studio Code. To explore AI-assisted code generation using GitHub Copilot. 	Week1 - wednesday	

- To analyze the accuracy and effectiveness of Copilot's code suggestions.
- To understand prompt-based programming using comments and code context

Lab Outcomes (LOs):

After completing this lab, students will be able to:

- Set up GitHub Copilot in VS Code successfully.
- Use inline comments and context to generate code with Copilot.
- Evaluate AI-generated code for correctness and readability.
- Compare code suggestions based on different prompts and programming styles.

Task Description#1

- Write a comment: # Function to check if a string is a valid palindrome (ignoring spaces and case) and allow Copilot to complete it.

Expected Output#1

- A function that correctly returns True for phrases like "A man a plan a canal Panama"

PROMPT-1

WRITE A CODE IN FUNCTION WHETHER THE STRING IS VALID PALINDROME OR NOT AND IGNORE THE SPACES AND CASE

The screenshot shows the Visual Studio Code interface with the Explorer, AI ASSISTED CODING, and TERMINAL panels. The AI ASSISTED CODING panel displays a prompt: "write a code in funtion whether the string is valid palindrome or not and ignore the spaces and case". Below the prompt, the generated Python code is shown:

```

1
2 def is_valid_palindrome(s):
3     cleaned = ''.join(c.lower() for c in s if c.isalnum())
4     return cleaned == cleaned[::-1]
5
6 # Example usage:
7 print(is_valid_palindrome("A man a plan a canal Panama")) # Should print True
8 print(is_valid_palindrome("Hello World")) # Should print False
9

```

The TERMINAL panel shows the output of the code execution:

```

on313/python.exe "c:/Users/priya/OneDrive/Desktop/AI ASSISTED CODING/lab1"
True
False
PS C:\Users\priya\OneDrive\Desktop\AI ASSISTED CODING
>

```

PROMPT-2

WRITE THE CODE WHETHER THE STRING IS VALID PALINDROME OR NOT AND IGNORE SPACES AND CASE BY USING FUNCTIONS

```
8 print(is_valid_palindrome("Hello World")) # Should print False
9
10 write the code whether the string is valid palindrome or not and ignore spaces and case by using functions
11
12 Ask Copilot
13
14 Accept Close
15
16 def check_palindrome():
17     s = input("Enter a string: ")
18     if is_valid_palindrome(s):
19         print("The string is a valid palindrome.")
20     else:
21         print("The string is not a valid palindrome.")
22
23 check_palindrome()
```

DEBUG CONSOLE

True
False
Enter a string: A man a plan a canal Panama
The string is a valid palindrome.
PS C:\Users\priya\OneDrive\Desktop\AI ASSISTED CODING
>

Task Description#2

- Generate a Python function that returns the Fibonacci sequence up to n terms. Prompt with only a function header and docstring

Expected Output#2

- AI completes the function logic using loop or recursion with accurate output

PROMPT-1

WRITE A CODE USING FUNCTIONS THAT RETURNS THE FIBONACCI SEQUENCE UP TO N TERMS ALONG WITH THE OUTPUT

```
1 def fibonacci_sequence(n):
2     write the code using function that returns the fibonacci sequence up to n terms along with the output
3
4     Ask Copilot
5
6     Accept Close
7
8     sequence = []
9     a, b = 0, 1
10    for _ in range(n):
11        sequence.append(a)
12        a, b = b, a + b
13    return sequence
14
15 # Example usage:
16 n = 10
17 result = fibonacci_sequence(n)
```

DEBUG CONSOLE

on313/python.exe "c:/Users/priya/OneDrive/Desktop/AI ASSISTED CODING/lab1"
Fibonacci sequence up to 10 terms: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
PS C:\Users\priya\OneDrive\Desktop\AI ASSISTED CODING
>

PROMPT-2

WRITE THE CODE THAT RETURNS THE FIBONACCI SEQUENCE UP TO N TERMS BY USING FUNCTIONS

The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a project named 'lab1' with a file 'text.py'. The main editor area displays a Python function generated by Copilot: `def get_fibonacci_up_to_n_terms(n):` followed by `return fibonacci_sequence(n)`. Below the function definition is an example usage: `fib_seq = get_fibonacci_up_to_n_terms(n)` and `print("Fibonacci sequence using function:", fib_seq)`. The bottom panel shows the 'TERMINAL' tab with the output: 'Fibonacci sequence up to 10 terms: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]' and 'Fibonacci sequence using function: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]'. The path 'PS C:\Users\priya\OneDrive\Desktop\AI ASSISTED CODING' is also visible.

Task Description#3

- Write a comment like # Function to reverse a string and use Copilot to generate the function.

Expected Output#3

- Auto-completed reverse function

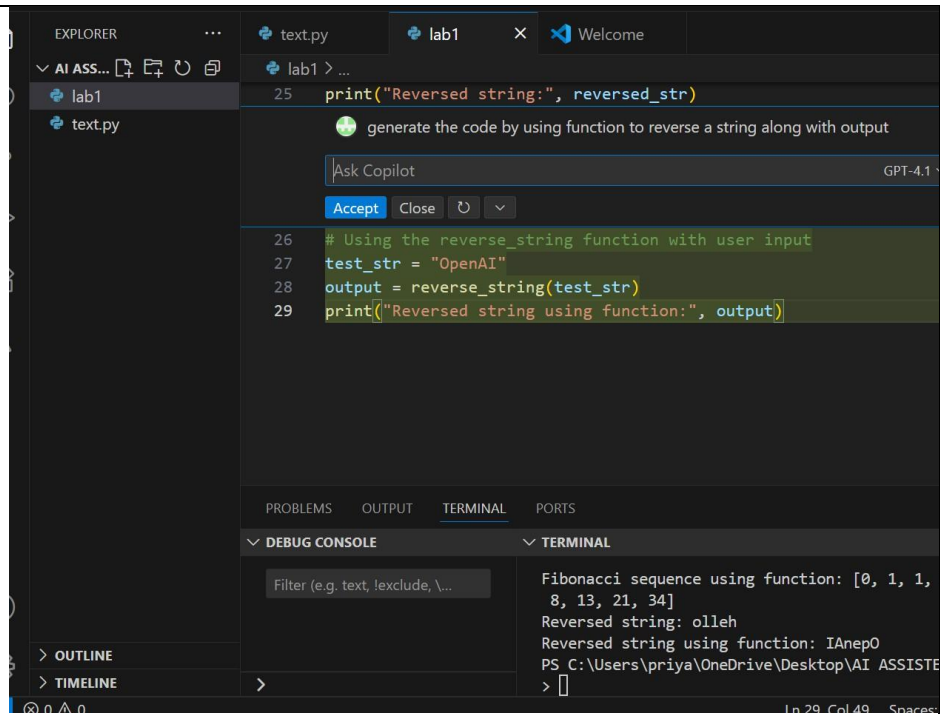
PROMPT-1

WRITE THE CODE USING FUNCTION TO REVERSE A STRING ALONG WITH OUTPUT

The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a project named 'lab1' with a file 'text.py'. The main editor area displays a Python function generated by Copilot: `def reverse_string(s):` followed by `return s[::-1]`. Below the function definition is an example usage: `input_str = "hello"`, `reversed_str = reverse_string(input_str)`, and `print("Reversed string:", reversed_str)`. The bottom panel shows the 'TERMINAL' tab with the output: '8, 13, 21, 34', 'Fibonacci sequence using function: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]', and 'Reversed string: olleh'. The path 'PS C:\Users\priya\OneDrive\Desktop\AI ASSISTED CODING' is also visible.

PROMPT-2

GENERATE THE CODE BY USING FUNCTION TO REVERSE A STRING ALONG WITH OUTPUT



```
25 print("Reversed string:", reversed_str)

generate the code by using function to reverse a string along with output

Ask Copilot GPT-4.1

Accept Close ↻ ▾

26 # Using the reverse_string function with user input
27 test_str = "OpenAI"
28 output = reverse_string(test_str)
29 print("Reversed string using function:", output)
```

PROBLEMS OUTPUT TERMINAL PORTS

DEBBUG CONSOLE TERMINAL

Filter (e.g. text, lexclude, \...

Fibonacci sequence using function: [0, 1, 1, 8, 13, 21, 34]
Reversed string: olleh
Reversed string using function: IAnepO
PS C:\Users\priya\OneDrive\Desktop\AI ASSISTED CODING
> █

Task Description#4

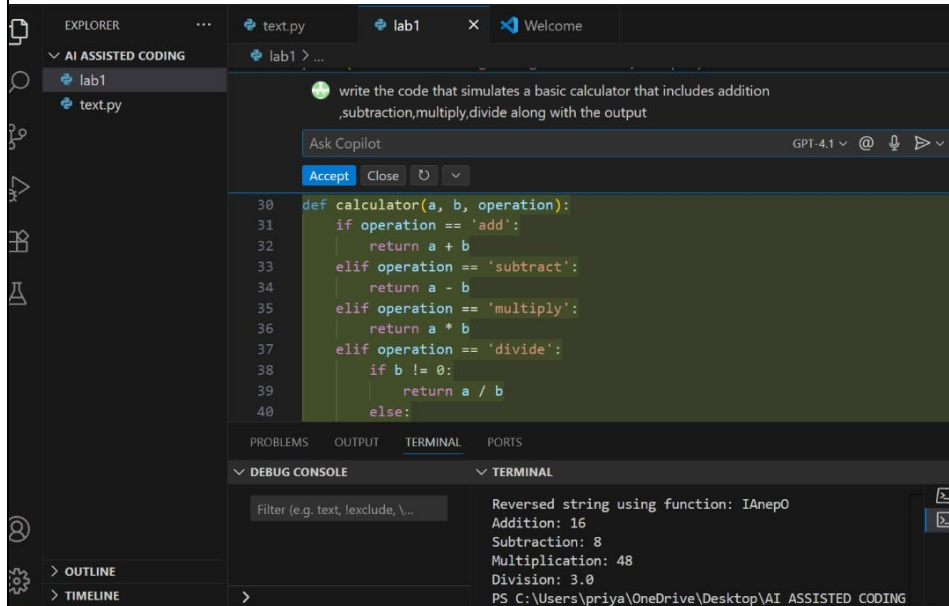
- Generate a program that simulates a basic calculator (add, subtract, multiply, divide).
Write the comment: # Simple calculator with 4 operations and let AI complete it.

Expected Output#4

- Fully working calculator with input/output and operator selection logic

PROMPT-1

WRITE THE CODE THAT SIMULATES A BASIC CALCULATOR THAT INCLUDES ADDITION ,SUBTRACTION,MULTIPLY,DIVIDE ALONG WITH OUTPUT



```
30 def calculator(a, b, operation):
31     if operation == 'add':
32         return a + b
33     elif operation == 'subtract':
34         return a - b
35     elif operation == 'multiply':
36         return a * b
37     elif operation == 'divide':
38         if b != 0:
39             return a / b
40     else:
```

PROBLEMS OUTPUT TERMINAL PORTS

DEBBUG CONSOLE TERMINAL

Filter (e.g. text, lexclude, \...

Reversed string using function: IAnepO
Addition: 16
Subtraction: 8
Multiplication: 48
Division: 3.0
PS C:\Users\priya\OneDrive\Desktop\AI ASSISTED CODING
> █

PROMPT-2

GENERATE A BASIC CALCULATOR WITH BASIC OPERATION SUCH AS ADDITION,SUBTRACTION,MULTIPLY,DIVISON

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying 'lab1' and 'text.py'. The main editor window shows a Python function `calculator(a, b, operation)` with comments and code for addition, subtraction, multiplication, and division. The AI Copilot interface is visible at the top, showing the prompt 'write the code that simulates a basic calculator that includes addition, subtraction, multiply, divide along with the output' and the 'Accept' button. The bottom panel shows the 'TERMINAL' tab with the output of the calculator function: 'Reversed string using function: IAnep0', 'Addition: 16', 'Subtraction: 8', 'Multiplication: 48', 'Division: 3.0', and the file path 'PS C:\Users\priya\OneDrive\Desktop\AI_ASSISTED CODING'.

```
def calculator(a, b, operation):
    if operation == 'add':
        return a + b
    elif operation == 'subtract':
        return a - b
    elif operation == 'multiply':
        return a * b
    elif operation == 'divide':
        if b != 0:
            return a / b
        else:
            return 0
```

Reversed string using function: IAnep0
Addition: 16
Subtraction: 8
Multiplication: 48
Division: 3.0
PS C:\Users\priya\OneDrive\Desktop\AI_ASSISTED CODING

Task Description#5

- Use a comment to instruct AI to write a function that reads a file and returns the number of lines..

Expected Output#5

- Functional implementation using `open()` or with `open()` and `readlines()`

PROMPT-1

WRITE A FUNCTION THAT READS A FILE AND RETURNS THE NUMBER OF LINES ALONG WITH THE OUTPUT

The screenshot shows the Google Colab interface with a Jupyter Notebook. The code cell contains a function `read_file_and_count_lines(file_path)` that reads a file and returns the number of lines and the file content. The AI assistant (Gemini) is shown in a chat window, providing a prompt to write a function that reads a file and returns the number of lines along with the output. The AI response includes a confirmation that it will write the function and a list of potential next steps, such as using the function with a specific file path.

```
def read_file_and_count_lines(file_path):
    """
    Reads a file, counts the lines, and returns the number of lines and the file content.
    """
    Args:
        file_path: The path to the file.

    Returns:
        A tuple containing:
        - The number of lines
        - A list of strings

    """
    line_count = 0
    lines = []
    try:
        with open(file_path, 'r') as file:
            for line in file:
                line_count += 1
                lines.append(line.strip())
    except FileNotFoundError:
        return 0, []
```

o write a function that reads a file and returns the number of lines along with the output

I can help with that! I'll write a function that reads a file, counts the lines, and returns both the line count and the content of the file.

The function `read_file_and_count_lines` is now defined and demonstrated with an example.

Here are some potential next steps you could take:

- Use the function with your own file: Replace `'my_file.txt'` with your own file path.

What can I help you build?

Untitled1.ipynb

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text ▶ Run all

```
# Create a dummy file for demonstration
with open('my_file.txt', 'w') as f:
    f.write("This is line 1.\n")
    f.write("This is line 2.\n")
    f.write("This is line 3.\n")

file_path = 'my_file.txt'
line_count, file_content = read_file_and_count_lines(file_path)

if line_count is not None and file_content is not None:
    print(f"The file '{file_path}' has {line_count} lines.")
    print("File content:")
    for i, line in enumerate(file_content):
        print(f"Line {i+1}: {line}")
```

The file 'my_file.txt' has 3 lines.
File content:
Line 1: This is line 1.
Line 2: This is line 2.
Line 3: This is line 3.

Variables Terminal 9:43 AM

PROMPT-2
WRITE THE CODE BY USING FUNCTIONS THAT READS A FILE AND RETURN THE NUMBER OF LINES

text.py lab1 2 x Welcome

lab1 > ...
23 print(f'(Using count_lines_in_file) total number of lines in {filename} : {lines

write the code by using functions that reads a file and returns the number of lines

Ask or edit in context GPT-4.1

Accept Close

```
24 def count_lines_v2(filename):
25     """
26     Reads a file and returns the number of lines in it (alternative implementation)
27     """
28     with open(filename, 'r', encoding='utf-8') as file:
29         lines = file.readlines()
```

PROBLEMS 2 OUTPUT TERMINAL PORTS

DEBUG CONSOLE

Filter (e.g. text, exclude, _)

PS C:\Users\priya\OneDrive\Desktop\AI ASSISTED CODING > & C:/Users/priya/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/priya/OneDrive/Desktop/AI ASSISTED CODING/lab1"

File "c:/Users/priya/OneDrive/Desktop/AI ASSISTED CODING/lab1", line 11
count = 0
AAAAA

IndentationError: expected an indented block after function definition on line 10

PS C:\Users\priya\OneDrive\Desktop\AI ASSISTED CODING >

powershell Python Python

incorrect, it returns None.

Finally, the code prints the

student_profile tuple and the first 10 Fibonacci numbers as an example of how to use the defined functions. This structure demonstrates basic Python programming concepts such as tuples, functions, conditionals, loops, and variable-length arguments.

Add Context...

lab1 x

Add context (#), external

G. v

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

Evaluation Criteria:

Criteria	Max Marks
Task #1	0.5
Task #2	0.5
Task #3	0.5
Task #4	0.5
Task #5	0.5
Total	2.5 Marks