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	**Applicableto Batches**	24CSBTB01 To 24CSBTB39
AssignmentNumber:1.2(Present assignment number)/**24**(Total number of assignments)			

Q.No.	**Question**	***ExpectedTime to complete***
1	Lab 1: Environment Setup – GitHub Copilot and VS Code Integration	Week1 - wednesday

Lab Objectives:

- To install and configure GitHub Copilot in Visual Studio Code.
- To explore AI-assisted code generation using GitHub Copilot.
- 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

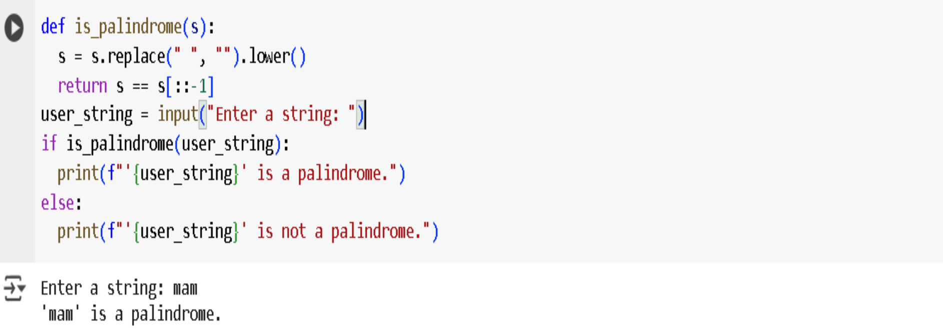
- Write a comment: # Function to check if a string is a valid palindrome (ignoring spaces and case) and allow Copilot to complete it.

Expected Output#1

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

PROMPT: Write a python program to check if a string is a valid palindrome ignoring spaces and case from the user.

CODE:



```
def is_palindrome(s):
  s = s.replace(" ", "").lower()
  return s == s[::-1]
user_string = input("Enter a string: ")
if is_palindrome(user_string):
  print(f"'{user_string}' is a palindrome.")
else:
  print(f"'{user_string}' is not a palindrome.")
```

```
Enter a string: mam
'mam' is a palindrome.
```

OBSERVATION: The string function is_palindrome that checks if a given string is a palindrome, ignoring spaces and case. It then prompts the user to enter a string, calls the function, and prints whether the entered string is a palindrome or not.

Task Description#2

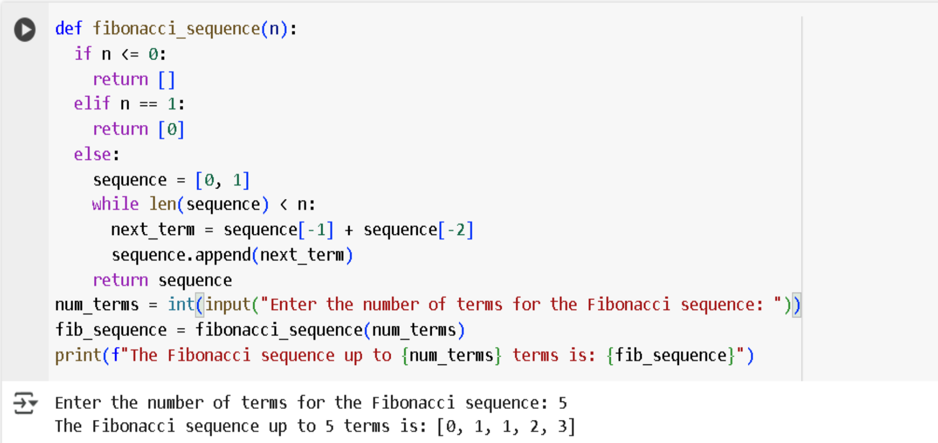
- Generate a Python function that returns the Fibonacci sequence up to n terms. Prompt with only a function header and docstring

Expected Output#2

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

PROMPT: Write a Python program that returns the Fibonacci sequence up to n terms

CODE:



```
def fibonacci_sequence(n):
  if n <= 0:
    return []
  elif n == 1:
    return [0]
  else:
    sequence = [0, 1]
    while len(sequence) < n:
      next_term = sequence[-1] + sequence[-2]
      sequence.append(next_term)
    return sequence
num_terms = int(input("Enter the number of terms for the Fibonacci sequence: "))
fib_sequence = fibonacci_sequence(num_terms)
print(f"The Fibonacci sequence up to {num_terms} terms is: {fib_sequence}")
```

```
Enter the number of terms for the Fibonacci sequence: 5
The Fibonacci sequence up to 5 terms is: [0, 1, 1, 2, 3]
```

OBSERVATION: This code defines a function fibonacci_sequence that generates the Fibonacci sequence up to a specified number of terms n. It handles edge cases for n <= 0 and n == 1, and for n > 1 it iteratively builds the sequence by adding the last two terms. The code then prompts the user to enter the number of terms, calls the function, and prints the resulting Fibonacci sequence.

Task Description#3

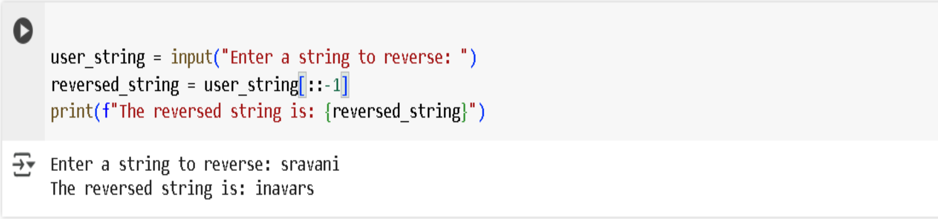
- Write a comment like # Function to reverse a string and use Copilot to generate the function.

Expected Output#3

Auto-completed reverse function

PROMPT: Write a python program to reverse a string take input from the user.

CODE:



```
user_string = input("Enter a string to reverse: ")
reversed_string = user_string[::-1]
print(f"The reversed string is: {reversed_string}")
```

```
Enter a string to reverse: sravani
The reversed string is: inavars
```

OBSERVATION: This code snippet takes a string input from the user using the input() function. It then reverses the string using slicing [::-1] which creates a reversed copy of the string. Finally, it prints the original string and the reversed string using an f-string for formatting.

Task Description#4

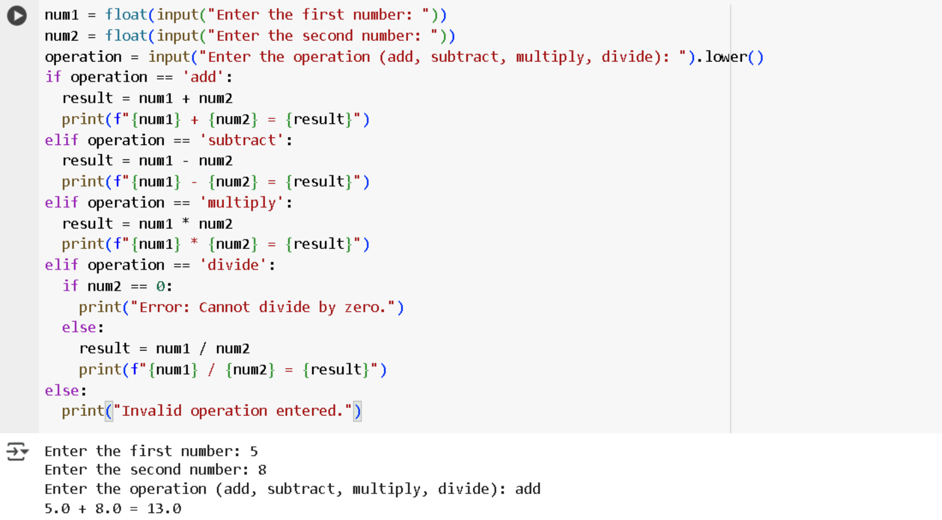
- Generate a program that simulates a basic calculator (add, subtract, multiply, divide). Write the comment: # Simple calculator with 4 operations and let AI complete it.

Expected Output#4

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

PROMPT: Write a program that simulates a basic calculator add, subtract, multiply, divide.

CODE:



```python
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
operation = input("Enter the operation (add, subtract, multiply, divide): ").lower()
if operation == 'add':
  result = num1 + num2
  print(f"{num1} + {num2} = {result}")
elif operation == 'subtract':
  result = num1 - num2
  print(f"{num1} - {num2} = {result}")
elif operation == 'multiply':
  result = num1 * num2
  print(f"{num1} * {num2} = {result}")
elif operation == 'divide':
  if num2 == 0:
    print("Error: Cannot divide by zero.")
  else:
    result = num1 / num2
    print(f"{num1} / {num2} = {result}")
else:
  print("Invalid operation entered.")
```

```
Enter the first number: 5
Enter the second number: 8
Enter the operation (add, subtract, multiply, divide): add
5.0 + 8.0 = 13.0
```

OBSERVATION:

This code implements a simple calculator. It prompts the user to enter two numbers and an operation (add, subtract, multiply, or divide). It then performs the selected operation on the numbers and prints the result. It also includes a check to prevent division by zero. If an invalid operation is entered, it prints an error message**.**

Task Description#5

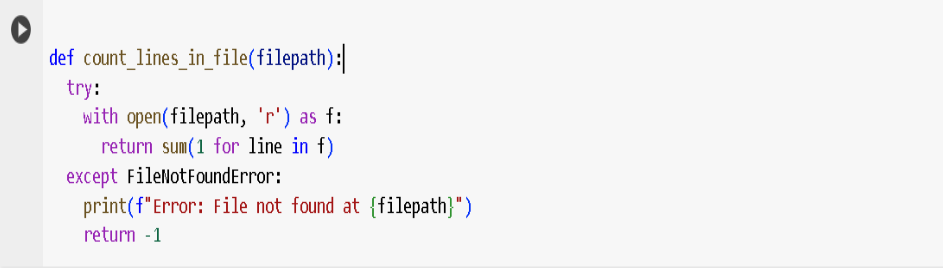
- Use a comment to instruct AI to write a function that reads a file and returns the number of lines..

Expected Output#5

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

PROMPT: Write a function that reads a file and returns the number of lines.

CODE:



```python
def count_lines_in_file(filepath):
  try:
    with open(filepath, 'r') as f:
      return sum(1 for line in f)
  except FileNotFoundError:
    print(f"Error: File not found at {filepath}")
    return -1
```

OBSERVATION:
The function attempts to open and read a file, counts the lines, and returns the count. If the file is not found, it prints an error and returns -1.

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**		