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 **Lab Objectives:** ● 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"

```python
# Function to check if a string is a valid palindrome
def is_palindrome(s):
    s = ''.join(c.lower() for c in s if c.isalnum())
    return s == s[::-1]

# Example usage and output
test_str = "A man, a plan, a canal: Panama"
print(is_palindrome(test_str))  # Output: True
```

OUTPUT:

```
"A man, a plan, a canal: Panama" is a palindrome: True
```

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

```python
def fibonacci_sequence(n):
    sequence = []
    a, b = 0, 1
    for _ in range(n):
        sequence.append(a)
        a, b = b, a + b
    return sequence

# Example usage and output
n = 10
result = fibonacci_sequence(n)
print(f"Fibonacci sequence up to {n} terms:")
print(result)
```

OUTPUT:

```
-ibonacci sequence up to 10 terms:
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

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

Untitled-4.py > ...

```python
def reverse_string(s):
    return s[::-1]

# Example usage
input_str = "hello"
output_str = reverse_string(input_str)
print("Reversed string:", output_str)
# Output: Reversed string: olleh
```

OUTPUT:

```
Reversed string: olleh
```

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

Untitled-5.py > ...

```python
def add(a, b):
    return a + b

def sub(a, b):
    return a - b

def multi(a, b):
    return a * b

def divide(a, b):
    if b == 0:
        return "Error: Division by zero"
    return a / b

# Example usage
x = 10
y = 5

print("Addition:", add(x, y))
print("Subtraction:", sub(x, y))
print("Multiplication:", multi(x, y))
print("Division:", divide(x, y))
```

OUTPUT:

```
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0
```

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()

```
def count_lines_in_file(file_path):

    try:
        with open('example.txt', 'r') as file:
            return sum(1 for line in file)
    except FileNotFoundError:
        print(f"The file at {file_path} does not exist.")
        return 0
    except Exception as e:
        print(f"An error occurred: {e}")
        return 0
# Example usage:
if __name__ == "__main__":
    file_path = 'example.txt'  # Replace with your file path
    line_count = count_lines_in_file(file_path)
    print(f"The file contains {line_count} lines.")
```

OUTPUT:

```
The file contains 2 lines.
```

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**