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	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	**Course Title**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	**Week1 - Monday**	**Time(s)**	24CSBTB01 To 24CSBTB39
Duration	2 Hours	**Applicable to Batches**	All batches
Assignment Number:1.1(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:**	Week1 - Monday

- 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

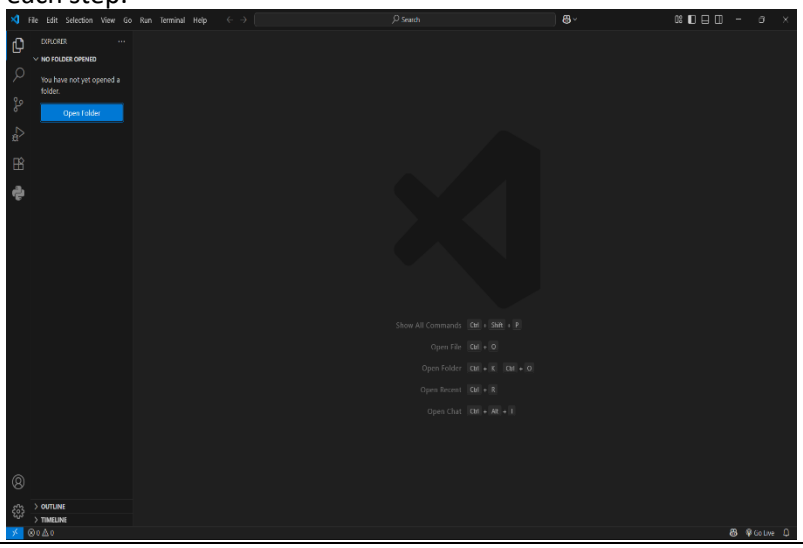
Lab Outcomes (LOs):
After completing this lab, students will be able to:

- 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 0

- Install and configure GitHub Copilot in VS Code. Take screenshots of each step.

Expected Output

- Install and configure GitHub Copilot in VS Code. Take screenshots of each step.
- 

Task 1: Factorial without Functions

- Description:
  Use GitHub Copilot to generate a Python program that calculates the factorial of a number without defining any functions (using loops directly in the main code).
- Expected Output:
  - A working program that correctly calculates the factorial for user-provided input.
  - Screenshots of the code generation process.

Prompt:

- create a python code that calculates the factorial of a number without defining any functions.

Code:

```
num = 5
factorial = 1
for i in range(1, num + 1):
    factorial *= i
print(factorial)
```

Output:

```
PS C:\PROGRAMMES VSCODE\AI coding>
PS C:\PROGRAMMES VSCODE\AI coding>  c:; cd 'c:\PROGRAMMES VSCODE\AI coding'; & 'c:\Users\venkatesh\AppDa
ions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '50128' '--' 'c:\PROGRAMMES VS
120
```

Task 2: Improving Efficiency

- Description:
  Examine the Copilot-generated code from Task 1 and demonstrate how its efficiency can be improved (e.g., removing unnecessary variables, optimizing loops).
- Expected Output:
  - Original and improved versions of the code.
  - Explanation of how the improvements enhance performance.

Prompt:

- remove the unnecessary variables, optimizing loops to improve the code.

Code:

```
num = 5
for i in range(1, num + 1):
    factorial *= i
print(factorial)
```

Output:

```
120
14400
PS C:\PROGRAMMES VSCODE\AI coding>
```

Task 3: Factorial with Functions

- Description:
  Use GitHub Copilot to generate a Python program that calculates the factorial of a number using a user-defined function.
- Expected Output:
  - Correctly working factorial function with sample outputs.
  - Documentation of the steps Copilot followed to generate the function.

Prompt:

- generate a Python program that calculates the factorial of a number using a user-defined function

Code:

```python
def factorial(n):
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result

print(factorial(5))
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\PROGRAMMES VSCODE\AI coding> & 'c:\Users\venkatesh\AppData\Local\Programs\Python\Python313\python.exe' 'c:\u
.0-win32-x64\bundled\libs\debugpy\launcher' '50944' '--' 'c:\PROGRAMMES VSCODE\AI coding\assignment 1.1task3.py'
120
PS C:\PROGRAMMES VSCODE\AI coding>
```

Comment:

- The code defines a function called factorial that calculates the factorial of a given number n.
- It uses a for loop to multiply all integers from 1 to n and returns the result.
- The function is then called with the argument 5, and the result is printed.

Task 4: Comparative Analysis – With vs Without Functions

- Description:
  Differentiate between the Copilot-generated factorial program with functions and without functions in terms of logic, reusability, and execution.
- Expected Output:

- A comparison table or short report explaining the differences.

Short Report:

- 1. Logic:
- With Functions: The logic is encapsulated within the `factorial` function, making it easier to understand and maintain.
- Without Functions: The logic is spread throughout the code, making it harder to follow and more prone to errors.
-
- 2. Reusability:
- With Functions: The `factorial` function can be reused with different inputs without duplicating code.
- Without Functions: The code is not reusable, as the factorial calculation is hardcoded.
-
- 3. Execution:
- - With Functions: The function is called with a specific input, and the result is returned, making it more efficient.
- - Without Functions: The code executes the factorial calculation directly, which can lead to redundancy and inefficiency.
-

Task 5: Iterative vs Recursive Factorial

- Description:
  Prompt GitHub Copilot to generate both iterative and recursive versions of the factorial function.
- Expected Output:
  - Two correct implementations.
  - A documented comparison of logic, performance, and execution flow between iterative and recursive approaches.

Prompt:

- Generate both iterative and recursive versions of the factorial function.

Code:

```python
def factorial_iterative(n):
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result

def factorial_recursive(n):
    if n == 0:
        return 1
    else:
        return n * factorial_recursive(n - 1)

print(factorial_iterative(5))
print(factorial_recursive(5))
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\PROGRAMMES VSCODE\AI coding>  & 'c:\Users\venkatesh\AppData\Local\Program
.0-win32-x64\bundled\libs\debugpy\launcher' '64897' '--' 'c:\PROGRAMMES VSCODE\
120
120
PS C:\PROGRAMMES VSCODE\AI coding>
```

Comment:

- The code defines a recursive function called factorial_recursive that calculates the factorial of a given number n.
- It uses a base case of n == 0 to return 1, and for all other values of n, it calls itself with n - 1.
- This effectively reduces the problem size with each call until it reaches the base case.

Submission Requirements

1. Generate code for each task with comments.
2. Screenshots of Copilot suggestions.
3. Comparative analysis reports (Task 4 and Task 5).
4. Sample inputs/outputs demonstrating correctness.

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
Successful Setup of Copilot	0.5

		Comparative Analysis – With vs Without Functions	1		
		Iterative vs Recursive Factorial	1		
		Total	**2.5 Marks**		