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:

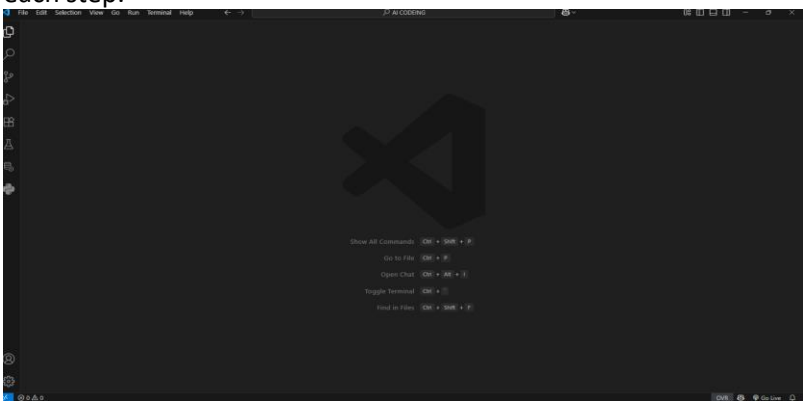
- Set up GitHub Copilot in VS Code successfully.
- Use inline comments and context to generate code with Copilot.
- Evaluate AI-generated code for correctness and readability.
- Compare code suggestions based on different prompts and programming styles.

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**:-
    # Factorial calculation without using function
  - **Code:-**

lab ass 1.1.py > ...

```python
# Factorial calculation without using functions

num = int(input("Enter a number: "))
factorial = 1

if num < 0:
    print("Factorial does not exist for negative numbers.")
elif num == 0:
    print("The factorial of 0 is 1.")
else:
    for i in range(1, num + 1):
        factorial *= i
    print(f"The factorial of {num} is {factorial}.")
```

  - **Output:-**

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL HISTORY TASK MONITOR

```
PS C:\AI CODEING>  & 'c:\Users\kbhuv\AppData\Local\Programs\Python\Python313\pyth
debugpy\launcher' '54816' '--' 'c:\AI CODEING\lab ass 1.1.py'
Enter a number: 4
The factorial of 4 is 24.
PS C:\AI CODEING>
```

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 :-**
    # This program calculates the factorial of a given number.
  - **CODE :-**

```
    num = int(input("Enter a number: "))

if num < 0:
    print("Factorial does not exist for negative numbers.")
else:
    result = 1
    for i in range(2, num + 1):
        result *= i
    print(f"The factorial of {num} is {result}.")
```

- **OutPut :-**

```
PS C:\AI CODEING>  c:; cd 'c:\AI CODEING'; & 'c:\Users\kbhuv\AppData\
-win32-x64\bundled\libs\debugpy\launcher' '52910' '--' 'c:\AI CODEING
Enter a number: 6
The factorial of 6 is 720.
Enter a number: 5
The factorial of 5 is 120.
```

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 :-**
    # Write a Python program that defines a function to calculate the factorial of a number.
    # The function should return None for negative numbers.
    # Ask the user to input a number, call the function with that input.
    # print the factorial if it's valid, or an error message if the number is negative.
  - **CODE :-**

```python
def factorial(n):
    if n < 0:
        return None
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result

num = int(input("Enter a number: "))

fact = factorial(num)

if fact is None:
    print("Factorial does not exist for negative numbers.")
else:
    print(f"The factorial of {num} is {fact}.")
```

- **OUTPUT :-**

```
PS C:\AI CODEING>  c:; cd 'c:\AI CODEING'; & 'c:\Use
-win32-x64\bundled\libs\debugpy\launcher' '62269' '--
Enter a number: 4
The factorial of 4 is 24.
```

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.
  - **PROMPT :-**
    # Factorial Program With a Function code
  - **CODE :-**

```python
#Factorial Program With a Function code
def factorial(n):
    if n < 0:
        return None
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result

num = int(input("Enter a number: "))
fact = factorial(num)

if fact is None:
    print("Factorial does not exist for negative numbers.")
else:
    print(f"The factorial of {num} is {fact}.")
```

  - **OUTPUT :-**

```
PS C:\AI CODEING>  c:; cd 'c:\AI CODEING'; & 'c:\U
-win32-x64\bundled\libs\debugpy\launcher' '53399'
Enter a number: 3
The factorial of 3 is 6.
```

- **PROMPT :-**
  # Factorial Program Without Functions Code:

- **CODE :-**

```python
#Factorial Program Without Functions Code:
num = int(input("Enter a number: "))

if num < 0:
    print("Factorial does not exist for negative numbers.")
else:
    result = 1
    for i in range(2, num + 1):
        result *= i
    print(f"The factorial of {num} is {result}.")
```

- **OUTPUT :-**

```
PS C:\AI CODEING>  c:; cd 'c:\AI CODEING'; & 'c
-win32-x64\bundled\libs\debugpy\launcher' '53456
Enter a number: 5
The factorial of 5 is 120.
```

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 :-**
    # Write a Python program that defines two functions to calculate the factorial of a number:
    # 1. An iterative version using a for loop.
    # 2. A recursive version that calls itself.
    # The program should ask the user to enter a number and display the result from both methods.
    # Handle negative inputs by showing an appropriate message.
  - **CODE :-**

```python
# Function to calculate factorial iteratively
def factorial_iterative(n):
    if n < 0:
        return None
    result = 1
    for i in range(2, n + 1):
        result *= i
    return result

def factorial_recursive(n):
    if n < 0:
        return None
    if n == 0 or n == 1:
        return 1
    return n * factorial_recursive(n - 1)

num = int(input("Enter a number: "))

iter_result = factorial_iterative(num)
recur_result = factorial_recursive(num)

if iter_result is None or recur_result is None:
    print("Factorial does not exist for negative numbers.")
else:
    print(f"Iterative: The factorial of {num} is {iter_result}.")
    print(f"Recursive: The factorial of {num} is {recur_result}.")
```

- **OUTPUT :-**

```
PS C:\AI CODEING>  c:; cd 'c:\AI CODEING'; & 'c:\Use
-win32-x64\bundled\libs\debugpy\launcher' '64726' '-
Enter a number: 5
Iterative: The factorial of 5 is 120.
Recursive: The factorial of 5 is 120.
```

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**