

# Assignment 1.1

Name : Aravind Reddy

Hall Ticket No : 2303a51027

Batch No : 01

Subject : AI Assistant Coding

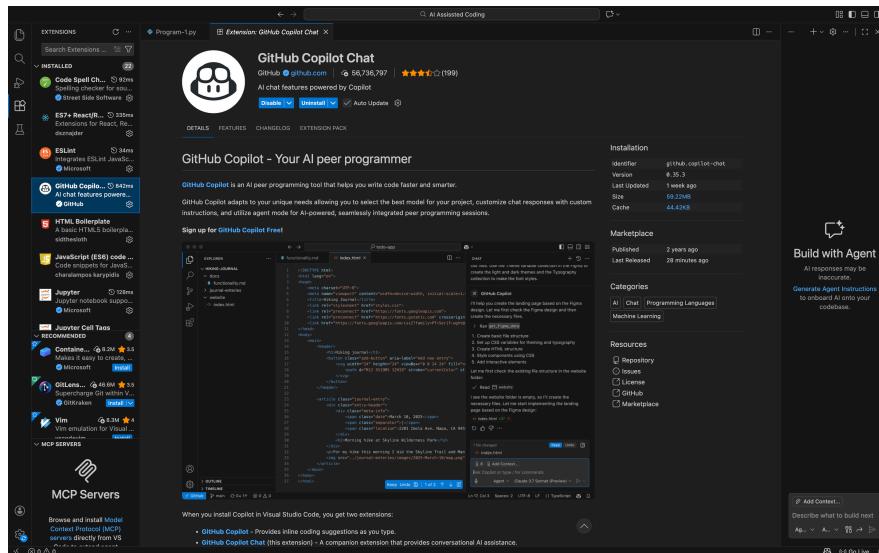
## Lab 1: Environment Setup – GitHub Copilot and VS Code Integration + Understanding

AI-assisted Coding Workflow.

### Task 0 :

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

### Expected Output



A screenshot of the Visual Studio Code interface. The title bar says "Walkthrough GitHub Copilot Chat". The left sidebar shows "EXPLORER", "SOURCE CONTROL - CHANGES", "AI ASSISTED CODING", "vscode", "Program-1.py", and "settings.json". The main editor area displays the following JSON code:

```
1 // editor.fontSize: 16,
2 // "editor.confirmDragAndDrop": false,
3 // "explorer.confirmDelete": false,
4 // "explorer.confirmRefresh": true,
5 // "editor.formatOnSave": true,
6 // "github.copilot.enable": {
7 //   "wt": true,
8 //   "plaintext": false,
9 //   "markdown": false,
10 //   "scmInput": false
11 // }
12 }
13 ]
14 }
```

The status bar at the bottom shows "Ln 1, Col 1 Spaces: 2 UTF-8 LF JSON with Comments 3.13.2 ⓘ Go Live ⌂ Prettier".

## Task 1: AI-Generated Logic Without Modularization (Factorial without Functions)

A screenshot of the Visual Studio Code interface. The title bar says "Assignment - 01.py". The left sidebar shows "EXPLORER", "SOURCE CONTROL - CHANGES", "AI ASSISTED CODING", "Assignment - 01.py", and "Program-1.py". The main editor area displays the following Python code:

```
1 # write a python program to calculate factorial of a number
2 # do not use functions
3 # use only loops and variables
4
5 number = int(input("Enter a number to calculate its factorial: "))
6 factorial = 1
7 for i in range(1, number + 1):
8     factorial *= i
9 print(f"The factorial of {number} is {factorial}")
10
11 # optimize this code and improve readability
12 # by adding comments and using meaningful variable names
13 # Get user input for the number
14 user_input_number = int(input("Enter a number to calculate its factorial: "))
15 # Initialize factorial result variable
16 factorial_result = 1
17 # Loop through numbers from 1 to user_input_number
18 for current_number in range(1, user_input_number + 1):
19     # Multiply the current number to the factorial result
20     factorial_result *= current_number
21 # Print the final factorial result
22 print(f"The factorial of {user_input_number} is {factorial_result}")
```

The terminal below shows the output of running the script:

```
/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignment - 01.py"
Enter a number to calculate its factorial: 5
The factorial of 5 is 120
% (base) = AI Assisted Coding [
```

The status bar at the bottom shows "Ln 7, Col 31 Spaces: 4 UTF-8 LF Python 3.13.2 ⓘ Go Live ⌂ Prettier".

## Task 2: AI Code Optimization & Cleanup (Improving Efficiency)

A screenshot of the Visual Studio Code interface. The title bar says "Assignment - 01.py". The left sidebar shows "EXPLORER", "SOURCE CONTROL - CHANGES", "AI ASSISTED CODING", "Assignment - 01.py", and "Program-1.py". The main editor area displays the following Python code:

```
1 # write a python program to calculate factorial of a number
2 # do not use functions
3 # use only loops and variables
4
5 number = int(input("Enter a number to calculate its factorial: "))
6 factorial = 1
7 for i in range(1, number + 1):
8     factorial *= i
9 print(f"The factorial of {number} is {factorial}")
10
11 # optimize this code and improve readability
12 # by adding comments and using meaningful variable names
13 # Get user input for the number
14 user_input_number = int(input("Enter a number to calculate its factorial: "))
15 # Initialize factorial result variable
16 factorial_result = 1
17 # Loop through numbers from 1 to user_input_number
18 for current_number in range(1, user_input_number + 1):
19     # Multiply the current number to the factorial result
20     factorial_result *= current_number
21 # Print the final factorial result
22 print(f"The factorial of {user_input_number} is {factorial_result}")
```

The terminal below shows the output of running the script:

```
/usr/local/bin/python3 "/Users/aravindreddy/Desktop/My-Information/College/AI Assisted Coding/Assignment - 01.py"
Enter a number to calculate its factorial: 5
The factorial of 5 is 120
% (base) = AI Assisted Coding [
```

The status bar at the bottom shows "Ln 10, Col 1 Spaces: 4 UTF-8 LF Python 3.13.2 ⓘ Go Live ⌂ Prettier".

## Task 3: Modular Design Using AI Assistance (Factorial with Functions)

```

1 # write a python program to calculate factorial using a function
2 # use meaningful variable names
3 # add comments
4 def calculate_factorial(n):
5     """Calculate the factorial of a given number n."""
6     # Initialize factorial result variable
7     factorial_result = 1
8     # Loop through numbers from 1 to n
9     for current_number in range(1, n + 1):
10        factorial_result *= current_number # Multiply the current number to the factorial result
11    return factorial_result # Return the final factorial result
12
13 n = int(input("Enter a number to calculate its factorial: ")) # Get user input
14 result = calculate_factorial(n) # Call the function to calculate factorial
15 print(f"The factorial of {n} is {result}.") # Print the result

```

The Factorial of 6 is 720.  
© (base) - AI Assisted Coding

## Task 4: Comparative Analysis – Procedural vs Modular AI Code (With vs Without Functions)

```

1 # Task 4: Comparative Analysis of a factorial calculation method - Procedural vs Modular AI Code (With vs
2 # Without Functions)
3
4 # Procedural Approach (Without Functions)
5 n = int(input("Enter a number to calculate its factorial: ")) # Get user input
6 factorial_result = 1 # Initialize factorial result variable
7 for current_number in range(1, n + 1):
8    factorial_result *= current_number # Multiply the current number to the factorial result
9 print(f"The factorial of {n} is {factorial_result}.") # Print the result
10
11 # In the procedural approach, the code is written in a linear fashion without using functions.
12 # This can lead to code duplication and makes it harder to maintain or reuse the code.
13 # Modular Approach (With Functions)
14
15 def calculate_factorial():
16     """Calculate the factorial of a given number n."""
17     factorial_result = 1 # Initialize factorial result variable
18     for current_number in range(1, n + 1):
19        factorial_result *= current_number # Multiply the current number to the factorial result
20    return factorial_result # Return the final factorial result
21
22 n = int(input("Enter a number to calculate its factorial: ")) # Get user input
23 result = calculate_factorial() # Call the function to calculate factorial
24 print(f"The factorial of {n} is {result}.") # Print the result
25
26 # In the modular approach, the code is organized into functions.
27 # This improves code readability, reusability, and maintainability.
28 # Functions allow us to encapsulate logic and make the code easier to test and debug.
29

```

The Factorial of 5 is 120.  
© (base) - AI Assisted Coding  
The Factorial of 5 is 120.  
© (base) - AI Assisted Coding  
The Factorial of 5 is 120.  
© (base) - AI Assisted Coding

## Task 5: AI-Generated Iterative vs Recursive Thinking

```

1 # Task 5: Iterative vs Recursive AI-Generated Code
2
3 # generate iterative factorial program in python
4 def iterative_factorial(n):
5     result = 1
6     for i in range(2, n + 1):
7         result *= i
8     return result
9 print("Iterative Factorial of 5:", iterative_factorial(5))
10
11 # generate recursive factorial program in python
12 def recursive_factorial(n):
13     if n == 0:
14         return 1
15     else:
16         return n * recursive_factorial(n-1)
17 print("Recursive Factorial of 5:", recursive_factorial(5))
18
19 # generate recursive factorial program in python
20 def recursive_factorial(n):
21     if n == 0:
22         return 1
23     else:
24         return n * recursive_factorial(n-1)
25 print("Recursive Factorial of 5:", recursive_factorial(5))

```

Iterative Factorial of 5: 120  
Recursive Factorial of 5: 120  
Recursive Factorial of 5: 120  
© (base) - AI Assisted Coding