SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL		**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**	
Name: Achina Manasa **Enroll Number: 2403A53043** **Batch No: 24BTCAICYB02** **ProgramName: B.Tech**		**Assignment Type: Lab**	**AcademicYear:**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	**CourseTitle**	AI Assisted Coding
Year/Sem	II/I	**Regulation**	R24
Date and Day of Assignment	Week1 - Thursday	**Time(s)**	
Duration	2 Hours	**Applicableto Batches**	24CSBTB01 To 24CSBTB39
AssignmentNumber:1.4(Present assignment number)/**24**(Total number of assignments)			

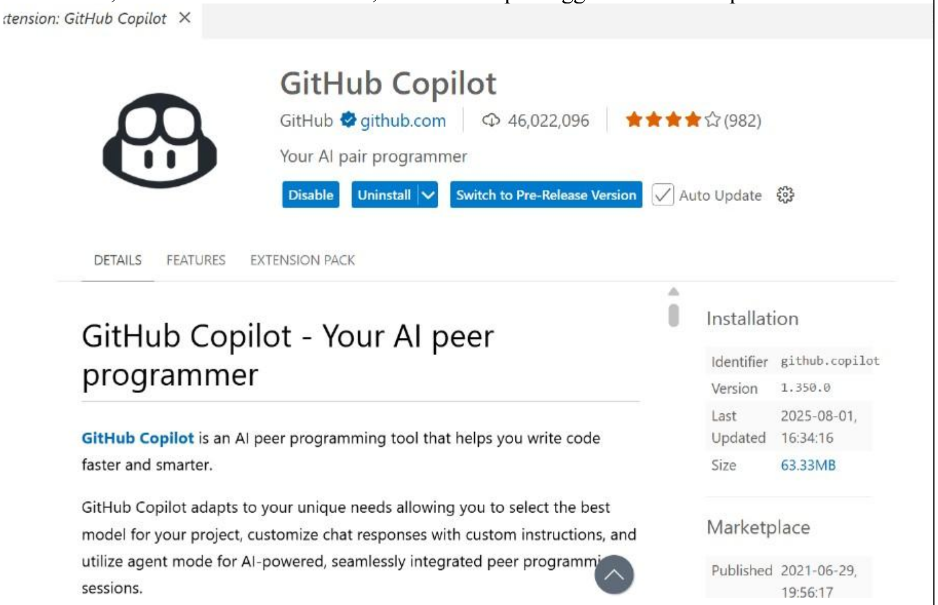
Q.No.	Question	*ExpectedTime to complete*
1	Lab 1: Environment Setup – GitHub Copilot and VS Code Integration **Lab Objectives:** ● To install and configure GitHub Copilot in Visual Studio Code. ● To explore AI-assisted code generation using GitHub Copilot.	Week1 - Thursday

- 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
• Install and configure GitHub Copilot in VS Code. Take screenshots of each step.
Expected Output #1
• Successfully install and activate GitHub Copilot in VS Code. Include screenshots showing installation, authentication via GitHub, and an example suggestion from Copilot**.**



GitHub Copilot

GitHub github.com | 46,022,096 | ★★★★☆ (982)

Your AI pair programmer

GitHub Copilot - Your AI peer programmer

GitHub Copilot is an AI peer programming tool that helps you write code faster and smarter.

GitHub Copilot adapts to your unique needs allowing you to select the best model for your project, customize chat responses with custom instructions, and utilize agent mode for AI-powered, seamlessly integrated peer programmi sessions.

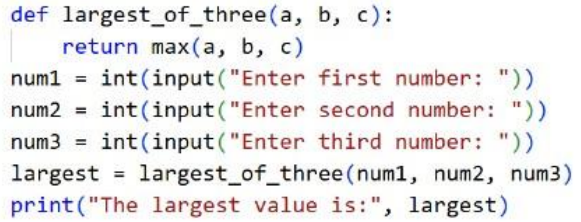
Installation

Identifier	github.copilot
Version	1.350.0
Last Updated	2025-08-01, 16:34:16
Size	63.33MB

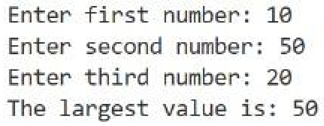
Marketplace

Published	2021-06-29, 19:56:17

Task Description #2
• A function in Python that returns the maximum of three numbers using GitHub Copilot. Use an appropriate comment as a prompt.
Expected Output #2
• Python function that takes three inputs and returns the largest value. Include the code and output.



```python
def largest_of_three(a, b, c):
    return max(a, b, c)
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
num3 = int(input("Enter third number: "))
largest = largest_of_three(num1, num2, num3)
print("The largest value is:", largest)
```



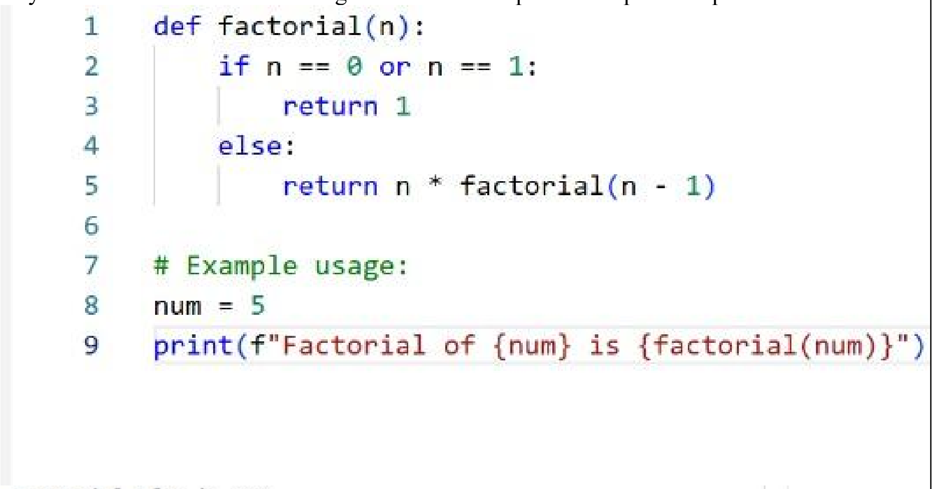
```
Enter first number: 10
Enter second number: 50
Enter third number: 20
The largest value is: 50
```

Task Description #3

• Use GitHub Copilot to create a recursive Python function that calculates the factorial of a number.

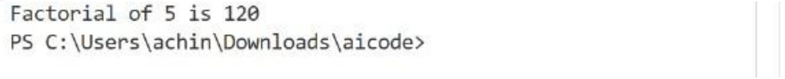
Expected Output #3

• Python function for factorial using recursion with input and output examples.



```python
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

# Example usage:
num = 5
print(f"Factorial of {num} is {factorial(num)}")
```



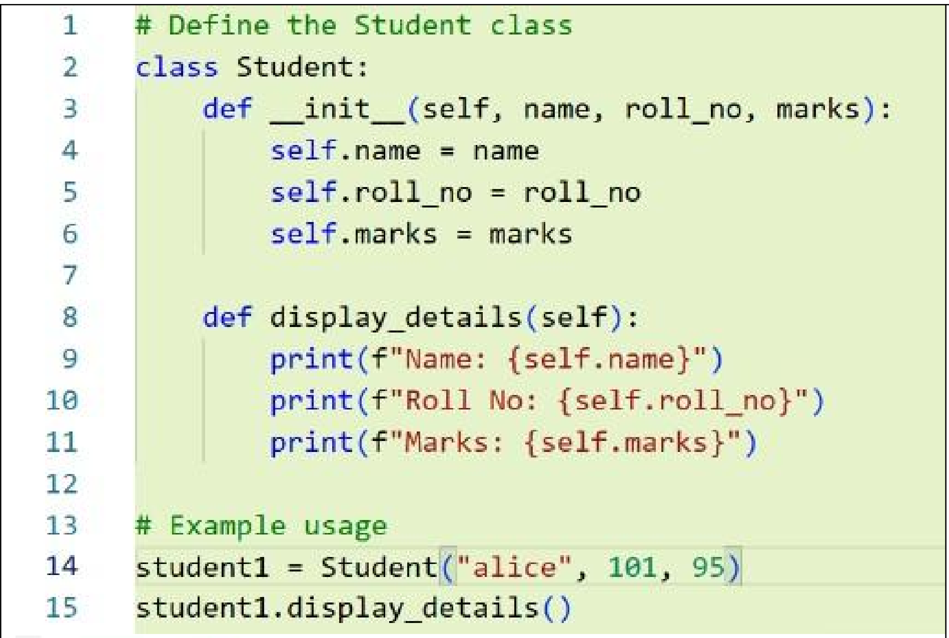
```
Factorial of 5 is 120
PS C:\Users\achin\Downloads\aicode>
```

Task Description #4

• Prompt GitHub Copilot to create a class named Student with attributes name, roll_no, and marks. Add a method to display student details.

Expected Output #4

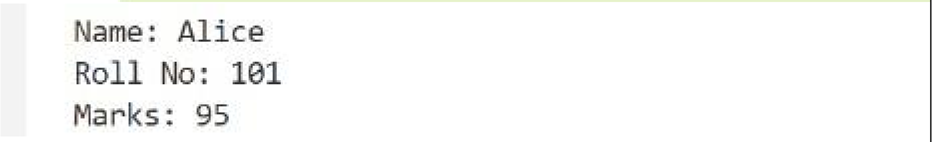
• Python class definition with an initializer and a display method. Include object creation and output.



```python
# Define the Student class
class Student:
    def __init__(self, name, roll_no, marks):
        self.name = name
        self.roll_no = roll_no
        self.marks = marks

    def display_details(self):
        print(f"Name: {self.name}")
        print(f"Roll No: {self.roll_no}")
        print(f"Marks: {self.marks}")

# Example usage
student1 = Student("alice", 101, 95)
student1.display_details()
```



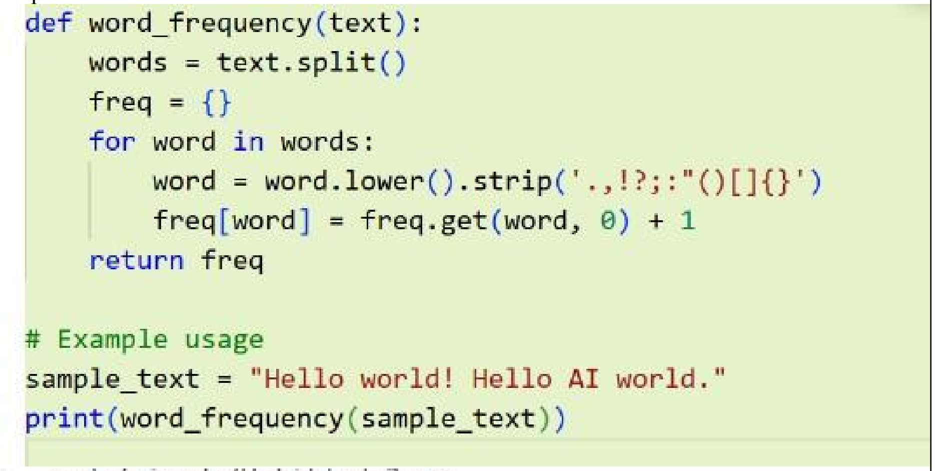
```
Name: Alice
Roll No: 101
Marks: 95
```

Task Description #5

• Ask GitHub Copilot to generate a Python function that takes a string as input and returns the frequency of each word.

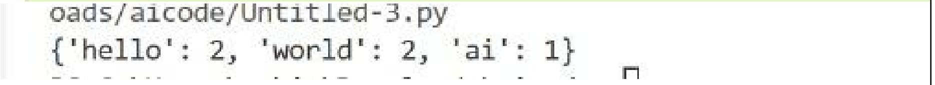
Expected Output #5

• Python function that returns word frequency using a dictionary. Provide sample input and output.



```python
def word_frequency(text):
    words = text.split()
    freq = {}
    for word in words:
        word = word.lower().strip('.,!?;:"()[]{}')
        freq[word] = freq.get(word, 0) + 1
    return freq

# Example usage
sample_text = "Hello world! Hello AI world."
print(word_frequency(sample_text))
```



```
oads/aicode/Untitled-3.py
{'hello': 2, 'world': 2, 'ai': 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
Install and configure GitHub Copilot in VS Code (Task #1)	0.5
Python function that takes three inputs and returns the largest value (Task #2)	0.5
Python function for factorial using recursion (Task #3)	0.5
Python class definition with an initializer and a display method (Task #4)	0.5
Function that returns word frequency using a dictionary (Task #5)	0.5
Total	**2.5 Marks**