SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
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**.**

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.

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.

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.

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.

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**		

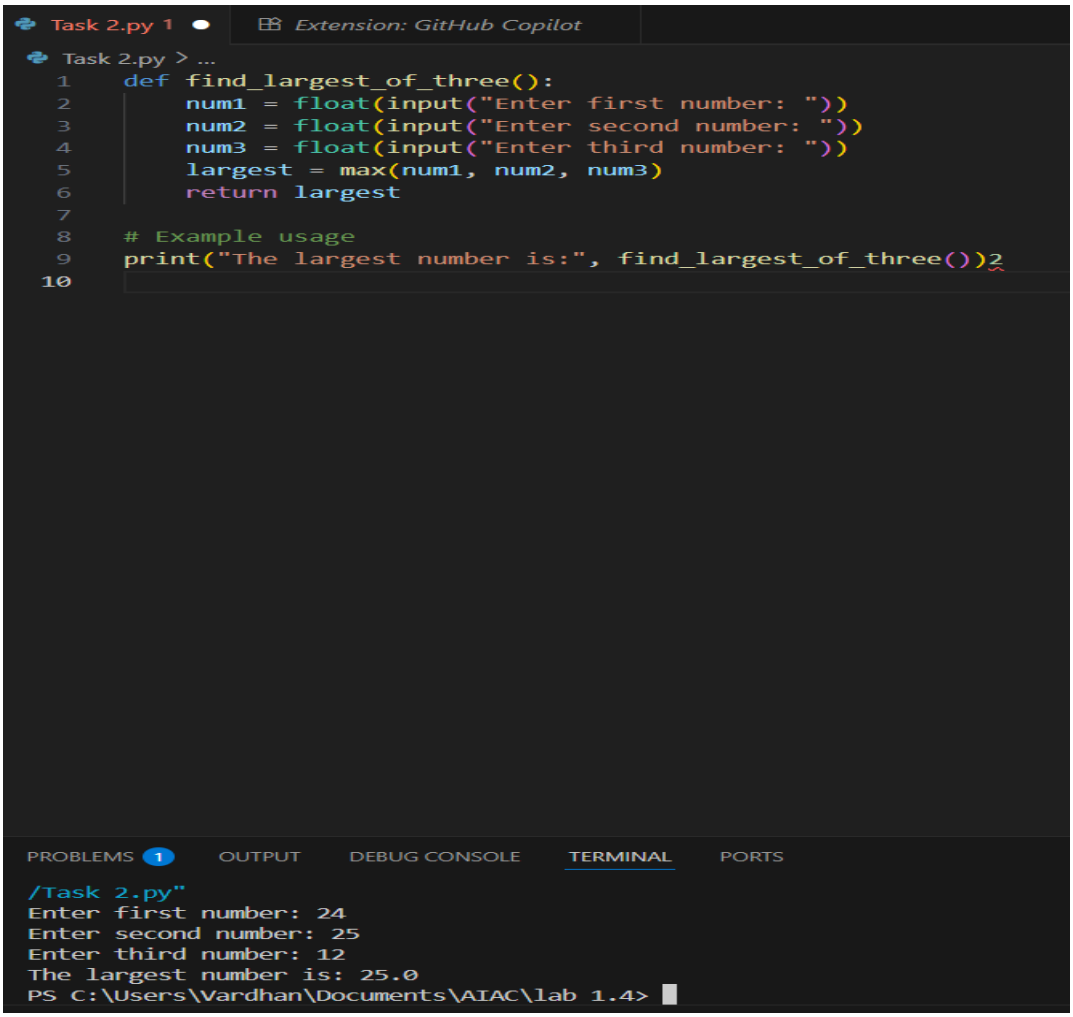
Task Description #2

• 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.

Prompt: create a function using python which compares the three numbers and returns the largest number by taking the input form the console.



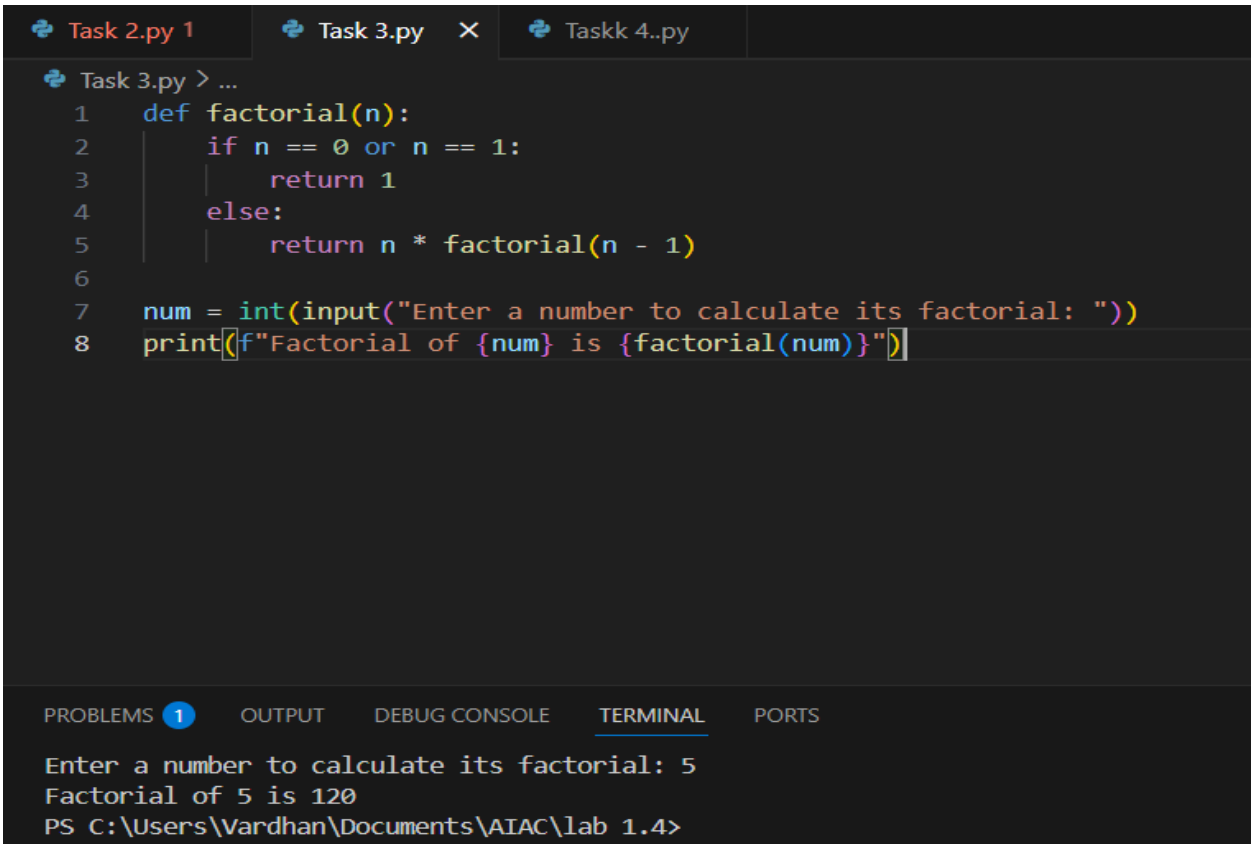
Task Description #3

• 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

Prompt:create a recursive python function which calculates the factorial of given a number by taking the input form the console.



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.

Prompt: create a class named Student with attributes name, roll_no, and marks. Add a method to display student details by taking input form the console for ‘n’ no.of student .
Python class definition with an initializer and a display method. Include object creation and output.

Taskk 4..py Task 2.py 1

Taskk 4..py > ...

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

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

students = []
n = int(input("Enter number of students: "))
for i in range(n):
    print(f"\nEnter details for student {i+1}:")
    name = input("Name: ")
    roll_no = input("Roll No: ")
    marks = float(input("Marks: "))
    student = Student(name, roll_no, marks)
    students.append(student)

print("\nStudent Details:")
for student in students:
    student.display()
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Vardhan\Documents\AIAC\lab 1.4>  & 'c:\Users\Vardhan\AppData\Local\Programs\Python\Python312\pyt
scode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '61046' '--' 'c:\Users
.4\Taskk 3..py'
Enter number of students: 1

Enter details for student 1:
Name: shiva
Roll No: 22
Marks: 80

Student Details:
Name: shiva, Roll No: 22, Marks: 80.0
PS C:\Users\Vardhan\Documents\AIAC\lab 1.4>
```

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.

Prompt: create a python function which takes a string as input form the console and return the frequency of each word using the a dictionary

Task 5.py > ...

```python
def word_frequency():
    text = input("Enter a string: ")
    words = text.split()
    freq = {}
    for word in words:
        word = word.lower()
        freq[word] = freq.get(word, 0) + 1
    return freq

# Example usage
if __name__ == "__main__":
    result = word_frequency()
    print(result)
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Enter a string: shiva
{'shiva': 1}
PS C:\Users\Vardhan\Documents\AIAC\lab 1.4> & C:/Users/Vardhan/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Vardhan/Documents/AIAC/lab 1.4/Task 5.py"
Enter a string: or
{'or': 1}
PS C:\Users\Vardhan\Documents\AIAC\lab 1.4> & C:/Users/Vardhan/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Vardhan/Documents/AIAC/lab 1.4/Task 5.py"
Enter a string: the
{'the': 1}
PS C:\Users\Vardhan\Documents\AIAC\lab 1.4> & C:/Users/Vardhan/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/Vardhan/Documents/AIAC/lab 1.4/Task 5.py"
Enter a string: rishi
{'rishi': 1}
PS C:\Users\Vardhan\Documents\AIAC\lab 1.4>
```