

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	Applicable to Batches	24CSBTB01 To 24CSBTB39
Assignment Number: 1.4 (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  <b>Lab Objectives:</b> <ul style="list-style-type: none"> <li>To install and configure GitHub Copilot in Visual Studio Code.</li> <li>To explore AI-assisted code generation using GitHub Copilot.</li> </ul>	Week1 - Thursday	

	<ul style="list-style-type: none"> <li>• To analyze the accuracy and effectiveness of Copilot's code suggestions.</li> <li>• To understand prompt-based programming using comments and code context</li> </ul> <p><b>Lab Outcomes (LOs):</b> After completing this lab, students will be able to:</p> <ul style="list-style-type: none"> <li>• Set up GitHub Copilot in VS Code successfully.</li> <li>• Use inline comments and context to generate code with Copilot.</li> <li>• Evaluate AI-generated code for correctness and readability.</li> <li>• Compare code suggestions based on different prompts and programming styles.</li> </ul> <p><b>Task Description #1</b></p> <ul style="list-style-type: none"> <li>• Install and configure GitHub Copilot in VS Code. Take screenshots of each step.</li> </ul> <p><b>Expected Output #1</b></p> <ul style="list-style-type: none"> <li>• Successfully install and activate GitHub Copilot in VS Code. Include screenshots showing installation, authentication via GitHub, and an example suggestion from Copilot.</li> </ul> <p><b>Task Description #2</b></p> <ul style="list-style-type: none"> <li>• A function in Python that returns the maximum of three numbers using GitHub Copilot. Use an appropriate comment as a prompt.</li> </ul> <p><b>Expected Output #2</b></p> <ul style="list-style-type: none"> <li>• Python function that takes three inputs and returns the largest value. Include the code and output.</li> </ul> <p><b>Task Description #3</b></p> <ul style="list-style-type: none"> <li>• Use GitHub Copilot to create a recursive Python function that calculates the factorial of a number.</li> </ul> <p><b>Expected Output #3</b></p> <ul style="list-style-type: none"> <li>• Python function for factorial using recursion with input and output examples.</li> </ul> <p><b>Task Description #4</b></p> <ul style="list-style-type: none"> <li>• Prompt GitHub Copilot to create a class named Student with attributes name, roll_no, and marks. Add a method to display student details.</li> </ul> <p><b>Expected Output #4</b></p> <ul style="list-style-type: none"> <li>• Python class definition with an initializer and a display method. Include object creation and output.</li> </ul> <p><b>Task Description #5</b></p> <ul style="list-style-type: none"> <li>• Ask GitHub Copilot to generate a Python function that takes a string as input and returns the frequency of each word.</li> </ul> <p><b>Expected Output #5</b></p> <ul style="list-style-type: none"> <li>• Python function that returns word frequency using a dictionary. Provide sample input and output.</li> </ul> <p><b>Note: Report should be submitted a word document for all tasks in a single document with prompts, comments &amp; code explanation, and output and if required, screenshots</b></p> <p><b>Evaluation Criteria:</b></p>	
--	--	--

	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		
	<b>Total</b>	<b>2.5 Marks</b>		

### 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: Write the maximum of given three numbers using python function

```

Task_2.py > ...
1  def maximum_of_three(a, b, c):
2      Click to collapse the range. , c)
3
4      # Example usage:
5      num1 = float(input("Enter first number: "))
6      num2 = float(input("Enter second number: "))
7      num3 = float(input("Enter third number: "))
8
9      print("The maximum number is:", maximum_of_three(num1, num2, num3))

```

```

PS C:\Users\solle\OneDrive\Desktop\AIAC\Lab -01> & C:/Users/solle/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/solle/OneDrive/Desktop/AIAC/Lab -01/Task_2.py"
Enter first number: 3
Enter second number: 5
Enter third number: 7
The maximum number is: 7.0

```

### 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 function using python that calculates factorial of a number

```

task_3.py > ...
1  def factorial(n):
2      if n < 0:
3          raise ValueError("Factorial is not defined for negative numb
4      if n == 0 or n == 1:
5          return 1
6      return n * factorial(n - 1)
7
8  try:
9      num = int(input("Enter a number to calculate its factorial: "))
10     print(f"Factorial of {num} is {factorial(num)}")
11 except ValueError as e:
12     print(f"Invalid input: {e}")

```

```

PS C:\Users\solle\OneDrive\Desktop\AIAC\Lab -01> & C:/Users/solle/A
ata/Local/Programs/Python/Python313/python.exe "c:/Users/solle/OneD
e/Desktop/AIAC/Lab -01/task_3.py"
Enter a number to calculate its factorial: 5
Factorial of 5 is 120

```

#### 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

task\_4.py × task\_3.py

task\_4.py > ...

```
1 class Student:
2     def __init__(self, name, roll_no, marks):
3         self.name = name
4         self.roll_no = roll_no
5         self.marks = marks
6
7     def display(self):
8         print(f"Name: {self.name}")
9         print(f"Roll No: {self.roll_no}")
10        print(f"Marks: {self.marks}")
11
12    students = []
13    n = int(input("Enter number of students: "))
14    for i in range(n):
15        print(f"\nEnter details for student {i+1}:")
16        name = input("Name: ")
17        roll_no = input("Roll No: ")
18        marks = float(input("Marks: "))
19        student = Student(name, roll_no, marks)
20        students.append(student)
21
22    print("\nStudent Details:")
23    for student in students:
24        student.display()
25        print("-" * 20)
```

```

PS C:\Users\solle\OneDrive\Desktop\AIAC\Lab -01> & C:/Users/solle/OneDrive/Desktop/
ata/Local/Programs/Python/Python313/python.exe "c:/Users/solle/OneDrive/Desktop/AIAC/Lab -01/task_4.py"
Enter number of students: 3

Enter details for student 1:
Name: siri
Roll No: 1
Marks: 20

Enter details for student 2:
Name: swadha
Roll No: 2
Marks: 19

Enter details for student 3:
Name: harshitha
Roll No: 3
Marks: 18

Student Details:
Name: siri
Roll No: 1
Marks: 20.0
-----
Name: swadha
Roll No: 2
Marks: 19.0
-----
Name: harshitha
Roll No: 3
Marks: 18.0

```

#### 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 from the console and return the frequency of each word using the a dictionary

task\_5.py X

task\_5.py > ...

```
1 def word_frequency():
```

C:\Users\solle\OneDrive\Desktop\AIAC\Lab -01\Lab\_Assignment\_1.4 (1).docx

```
3     words = text.split()
```

```
4     freq = {}
```

```
5     for word in words:
```

```
6         freq[word] = freq.get(word, 0) + 1
```

```
7     return freq
```

```
8
```

```
9     # Example usage:
```

```
10    # result = word_frequency()
```

```
11    # print(result)
```

Enter a string: ok not bad

{'ok': 1, 'not': 1, 'bad': 1}

PS C:\Users\Dattu\OneDrive\Documents\AIAS\Lab 1>