|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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. * 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** | | | | | | | Week1 - Thursday |  |

**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:

give me a python program that gives the maximum of three numbers in function format

Code:

def find\_max\_of\_three(a, b, c):

    return max(a, b, c)

# Example usage

num1 = 10

num2 = 20

num3 = 15

print("The maximum of the three numbers is:", find\_max\_of\_three(num1, num2, num3))

Output:

The maximum of the three numbers is: 20

def find\_max\_of\_three(a, b, c):

    return max(a, b, c)

# Example usage

num1 = 10

num2 = 20

num3 = 15

numbers = [num1, num2, num3]

print("The maximum of the three numbers is:", max(numbers))

**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:

def factorial(n):

    """

    Calculate the factorial of a number recursively.

    :param n: Non-negative integer

    :return: Factorial of n

    """

    if n < 0:

        raise ValueError("Factorial is not defined for negative numbers.")

    if n == 0 or n == 1:

        return 1

    return n \* factorial(n - 1)

# Example usage

if \_\_name\_\_ == "\_\_main\_\_":

    number = 5

    print(f"The factorial of {number} is {factorial(number)}")

Output:

The 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

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("John Doe", 101, 95)

# student1.display\_details()

Output:

student1 = Student("John Doe", 101, 95)

student1.display\_details()

**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:

generate a python function that takes a string as a input and returns the frequency of each word

def word\_frequency(text):

    """

    Takes a string input and returns the frequency of each word.

    Args:

        text (str): The input string.

    Returns:

        dict: A dictionary with words as keys and their frequencies as values.

    """

    words = text.split()

    frequency = {}

    for word in words:

        word = word.lower()  # Convert to lowercase for case-insensitive counting

      frequency[word] = frequency.get(word, 0) + 1

  return frequency