AI ASSISTED CODING

ASSIGNMENT-1.4

Name: S.SIRI

Hall Number:2403A51236

Batch:11

Task-1: A function in Python that returns the maximum of three numbers using GitHub Copilot. Use an appropriate comment as a prompt.

Expected Output -1:

• Python function that takes three inputs and returns the largest value. Include the code and output.

Prompt: Write a Python function that takes three numbers as input and returns the largest number among them. Include proper comments and an example usage.

Code:

```
# Prompt for GitHub Copilot: "Write a Python function that takes three numbers as input and returns
Tabnine | Edit | Test | Explain | Document
def max_of_three(a, b, c):
    """
    Returns the maximum of three numbers.
    Parameters:
    a (int or float): First number
    b (int or float): Second number
    c (int or float): Third number
    Returns:
    int or float: The largest of the three numbers
    """
    return max(a, b, c)  # Using Python's built-in max function
# Example usage
num1 = 10
num2 = 25
num3 = 15
print("The maximum number is:", max_of_three(num1, num2, num3))
```

Output:

```
sers/sonti/OneDrive/Documents/aiass2-9
The maximum number is: 25
PS C:\Users\sonti\OneDrive\Documents\a
```

Task-2: Use GitHub Copilot to create a recursive Python function that calculates the factorial of a
number.

Expected Output -2:

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

Prompt: Write a recursive Python function to calculate the factorial of a number. Include comments and an example showing input and output.

Code:

```python
# Prompt for GitHub Copilot: "Write a recursive Python function to calculate the factorial of a number with proper comments and exampl
Tabnine | Edit | Test | Explain | Document
def factorial(n):
    """
    Returns the factorial of a number using recursion.
    Parameters:
    n (int): Non-negative integer whose factorial is to be calculated
    Returns:
    int: Factorial of the input number
    Example:
    factorial(5) -> 120
    """
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)
# Example usage
num = 5
print(f"The factorial of {num} is:", factorial(num))
```

Output:

```
AI ASS-10.4/ass1.1/ass1.4/task-2"
The factorial of 5 is: 120
PS C:\Users\sonti\OneDrive\Documents\aiass2-9.2>
```

Task-3: 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-3:

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

Prompt: Create a Python class named Student with attributes name, roll_no, and marks. Add a method to display student details. Include example of object creation and output.

Code:

```
10.4 > ass1.1 > ass1.4 > task-3 > Student > display_details
# Prompt for GitHub Copilot: "Create a Python class named Student with attributes name, roll_no, a
class Student:
    """
    A class to represent a student.
    Attributes:
    name (str): Name of the student
    roll_no (int): Roll number of the student
    marks (float): Marks obtained by the student
    """
    Tabnine | Edit | Test | Explain | Document
    def __init__(self, name, roll_no, marks):
        self.name = name
        self.roll_no = roll_no
        self.marks = marks
    Tabnine | Edit | Test | Explain | Document
    def display_details(self):
        """Prints the details of the student"""
        print(f"Name: {self.name}")
        print(f"Roll Number: {self.roll_no}")
        print(f"Marks: {self.marks}")
# Example usage
student1 = Student("Alice", 101, 88.5)
student1.display_details()
```

Output:

```
sers/sonti/OneDrive/Documents/aiass2-9.2/AI
Name: Alice
Roll Number: 101
Marks: 88.5
PS C:\Users\sonti\OneDrive\Documents\aiass2
```

Task-4: Ask GitHub Copilot to generate a Python function that takes a string as input and returns the
frequency of each word.

Expected Output-4:

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

Prompt: Write a Python function that takes a string as input and returns the frequency of each word using a dictionary. Include sample input and output.

Code:

```
-10.4 > ass1.1 > ass1.4 > task-4 > ...
# Prompt for GitHub Copilot: "Write a Python function that takes a string as input and returns the
Tabnine | Edit | Test | Explain | Document
def word_frequency(text):
    """
    Returns the frequency of each word in the given string.
    Parameters:
    text (str): Input string
    Returns:
    dict: A dictionary where keys are words and values are their frequencies
    """
    words = text.split()
    frequency = {}
    for word in words:
        word = word.lower().strip(",.!?")  # normalize case and remove punctuation
        frequency[word] = frequency.get(word, 0) + 1
    return frequency
# Example usage
sample_text = "Hello world! Hello OpenAI world."
print("Word Frequency:", word_frequency(sample_text))
```

Output:

```
sers/sonti/OneDrive/Documents/aiass2-9.2/AI ASS-10.4/ass
Word Frequency: {'hello': 2, 'world': 2, 'openai': 1}
PS C:\Users\sonti\OneDrive\Documents\aiass2-9.2>
```