

# AI Assisted Coding

## Assignment-9.2

Name: N.Varshith

Ht.No: 2303A51441

Batch: 21

### TaskDescription-1(Documentation–FunctionSummary

Generation)

**Prompt:**

add a short functional summary as a docstring for every function in this file. The summary must explain the function purpose only, without describing logic or implementation steps.

**Code:**

```
...
Task Description -1 (Documentation ┌ Function Summary
Generation)
Task:
Use AI to generate concise functional summaries for each Python
function in a given script.
Instructions:
• Provide a Python script to the AI.
• Ask the AI to write a short summary describing the purpose of
each function.
• Ensure summaries are brief and technically accurate.
• Do not include code implementation details.
Expected Output -1:
A Python script where each function contains a clear and concise
summary explaining its purpose.''
# Sample Python script with function summaries
def calculate_area(radius):
    """
    Calculate the area of a circle given its radius.
    Parameters:
    radius (float): The radius of the circle.
    Returns:
    float: The area of the circle.
    """
    import math
    return math.pi * (radius ** 2)

print(calculate_area(10))
```

**Output:**

```
C:\Users\acer\\Desktop\Btech_3_2\AI Assistant coding>python
314.1592653589793
```

### **Observation:**

AI-generated function summaries help improve code readability by clearly describing the purpose of each function. The summaries make it easier for developers to understand the program structure without analyzing the implementation details. This approach also supports faster maintenance and documentation consistency.

### **Task Description-2 (Documentation–Logical Explanation for Conditions and Loops) Prompt:**

Add a brief explanation of the logic behind each condition and loop statements.  
Explain only decision - making and loop statements **Code:**

```
# Sample Python program with logical explanations for conditions and loops
def classify_number(num):
    """
    Classify a number as positive, negative, or zero.

    Parameters:
    num (int): The number to classify.

    Returns:
    str: A string indicating whether the number is positive, negative, or zero.
    """
    if num > 0:
        return "Positive"
    elif num < 0:
        return "Negative"
    else:
        return "Zero"
print(classify_number(-7))
def count_down(n):
    """
    Count down from a given number to zero.

    Parameters:
    n (int): The starting number for the countdown.

    Returns:
    list: A list of numbers counting down from n to 0.
    """
    countdown_list = []
    for i in range(n, -1, -1):
        countdown_list.append(i)
    return countdown_list
print(count_down(10))
```

### **Output:**

```
C:\Users\acer\Desktop\Btech_3_2\AI Assistant coding>python
Negative
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

### **Observation:**

AI-generated explanations for conditions and loops clarify the decision-making flow and iteration behavior within the program. These logical comments help users quickly understand how the program controls execution paths. As a result, debugging and code comprehension become more efficient.

### **Task Description-3(Documentation–File-Level-Overview) Prompt:**

Generate a file summarizing the overall functionality of the script, including the purpose of the main function and how it interacts with the user.

#### **Code:**

```
# Sample Python file with a file-level overview
"""
This Python file contains functions for performing basic mathematical operations such as addition, subtraction, multiplication, and division.
Each function takes two numerical inputs and returns the result of the respective operation.
The file also includes error handling for division by zero in the division function.
Additionally, there are test cases to validate the correctness of each function.
"""

class Calculator:
    def add(self, a, b):
        return a + b

    def subtract(self, a, b):
        return a - b

    def multiply(self, a, b):
        return a * b

    def divide(self, a, b):
        if b == 0:
            raise ValueError("Cannot divide by zero")
        return a / b

print("Sum:", Calculator().add(5, 3))
print("Difference:", Calculator().subtract(10, 4))
print("Product:", Calculator().multiply(6, 7))
print("Quotient:", Calculator().divide(20, 5))
```

#### **Output:**

```
C:\Users\acer\Desktop\Btech_3_2\AI Assistant coding>python
Sum: 8
Difference: 6
Product: 42
Quotient: 4.0
```

#### **Script Summary:**

This script defines a function called add, subtract, multiply, divide calculates the mathematical operations. The script also checks for an error occurred when divided by zero it throws the error “Value Error”.

#### **Observation:**

The AI successfully generated a concise file-level overview summarizing the overall purpose and functionality of the Python program. The overview was placed at the beginning of the file and described the program at a conceptual level

without including implementation details. This improved the readability of the code and helped users quickly understand the program's objective.

### TaskDescription-4(Documentation–Refine Existing Documentation)

#### Prompt:

Rewrite the existing comments to improve clarity and consistency while keeping the original meaning unchanged. Do not modify the program logic.

#### Code:

```
'''Task Description -4 (Documentation - Refine Existing Documentation)
Task:
Use AI to improve clarity and consistency of existing documentation in Python code.
Instructions:
• Provide Python code containing basic or unclear comments.
• Ask AI to rewrite the documentation to improve clarity and consistency.
• Ensure technical meaning remains unchanged.
Expected Output -4:
Python code with refined and improved documentation that is clear and consistent.'''
# Sample Python code with refined documentation
def calculate_area(radius):
    """
    Calculate the area of a circle based on its radius.
    Parameters:
    radius (float): The radius of the circle for which the area is to be calculated.

    Returns:
    float: The computed area of the circle using the formula A = π * r^2.
    """
    import math
    return math.pi * (radius ** 2)
```

#### Output:

```
C:\Users\acer\Desktop\Btech_3_2\AI Assistant coding>python
314.1592653589793
```

**Observation:** The script proved the clarity and consistency of the existing comments without altering the program's functionality. The refined documentation provided clearer explanations while preserving the original technical meaning. This enhancement made the code easier to read, understand, and maintain.

## TaskDescription-5(Documentation–Prompt Detail Impact Study) Brief

### Prompt:

Write a short one-line documentation comment describing the purpose of the following function.

### Code:

```
'''Write a short one-line documentation comment describing the purpose of the following function.'''
def factorial(n):
    """
    Returns int: The factorial of n, denoted as n!, which is the product of all positive integers less than or equal to n.
    """
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)
```

### Output:

```
C:\Users\acer\Desktop\Btech_3_2\AI Assistant coding>python
120
```

### Detailed Prompt:

Write a clear 3–5 line documentation comment for the following function describing its purpose, inputs, outputs, and overall behavior.

### Code:

```
'''Write a clear 3-5 line documentation comment for the following function describing its purpose, inputs, outputs, and overall behavior.'''
def factorial(n):
    """
    This function calculates the factorial of a given non-negative integer n. The factorial of n (denoted as n!) is the product of all positive integers less than or equal to n. The function uses recursion to compute the factorial, where the base case is defined for n = 0 and n = 1, returning 1 in both cases. For values of n greater than 1, the function calls itself with the argument n-1 and multiplies it by n to compute the result.
    Parameters:
    n (int): A non-negative integer for which the factorial is to be calculated.
    Returns:
    int: The factorial of the input integer n.
    """
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)
```

### Output:

```
C:\Users\acer\Desktop\Btech_3_2\AI Assistant coding>python
120
```

## **Observation:**

<b>Criteria</b>	<b>BriefPromptOutput(One-lineComment)</b>	<b>DetailedPromptOutput(3–5line Docstring)</b>
<b>Completeness</b>	Provides only the basic purpose of the function.	Includes purpose, input description, output, and behavior details.
<b>Clarity</b>	Clear but very minimal explanation.	More descriptive, structured, and easier to understand.
<b>Accuracy</b>	Correct but limited technical information.	Highly accurate with full technical context and parameter details.