ASSIGNMENT-9

Roll no:2303A52074
Batch-37

Task Description -1 (Documentation – Function Summary Generation)

Prompt

Generate a Python script containing 3 to 4 simple functions related to basic mathematics user defined input.
Write a short and technically accurate functional summary for each function in the following Python script.

Code

```python
#Generate a Python script containing 3 to 4 simple functions related to basic mathematics user defined input
#Function to add two numbers
def add(a, b):
    return a + b
#Function to subtract two numbers
def subtract(a, b):
    return a - b
#Function to multiply two numbers
def multiply(a, b):
    return a * b
#Function to divide two numbers
def divide(a, b):
    if b != 0:
        return a / b
    else:
        return "Cannot divide by zero"
#User input
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
#Perform operations
print("Addition:", add(num1, num2))
print("Subtraction:", subtract(num1, num2))
print("Multiplication:", multiply(num1, num2))
print("Division:", divide(num1, num2))

#Write a short and technically accurate functional summary for each function in the following Python script.

#Function add(a, b): Adds two numbers a and b and returns their sum.
#Function subtract(a, b): Subtracts number b from number a and returns the result.
#Function multiply(a, b): Multiplies two numbers a and b and returns their product.
#Function divide(a, b): Divides number a by number b and returns the quotient. If b is zero, it returns an error message.
```

Output

```
Enter first number: 32
Enter second number: 12
Addition: 44.0
Subtraction: 20.0
Multiplication: 384.0
Division: 2.6666666666666665
```

Explanation

First, a Python script with functions is generated using AI.Then, AI is used again to create short and clear summaries for each function.
This shows how AI can help in automatic documentation generation.

Task Description -2 (Documentation – Logical Explanation for Conditions and Loops)

Prompt

Generate a simple Python program that includes conditional statements and loops.Do not include any comments

Explain the logic of the conditions and loops in the following Python program.Focus only on how decisions are made and how the loop works.

Code

```
#Generate a simple Python program that includes conditional statements and loops.Do not include any comments or documentation.
#User input
age = int(input("Enter age: "))
income = int(input("Enter income: "))
#Evaluate scholarship eligibility
if age >= 18:
    if income <= 300000:
        print("Eligible")
    else:
        print("Return Conditionally Eligible")
else:
    print("Not Eligible")

#Explain the logic of the conditions and loops in the following Python program.Focus only on how decisions are made and how the loop works.
#The program first takes user input for age and income.
# It then uses a nested conditional statement to evaluate scholarship eligibility.
# The outer condition checks if the age is greater than or equal to 18. If this condition is true,
# it proceeds to check the income. If the income is less than or equal to 300000,
# it prints "Eligible". If the income is greater than 300000, it prints "Return Conditionally Eligible".
# If the age condition is false (i.e., age is less than 18), it directly prints "Not Eligible".
# There are no loops in this program; it simply evaluates the conditions once based on the user input.
```

Output

```
M-6/AI/docs/codes/ai 9.py
Enter age: 18
Enter income: 3000000
Return Conditionally Eligible
#Explain the logic of the conditions and loops in the following Python program.Focus only on how decisions are made and how the loop works.
#The program first takes user input for age and income.
# It then uses a nested conditional statement to evaluate scholarship eligibility.
# The outer condition checks if the age is greater than or equal to 18. If this condition is true,
# it proceeds to check the income. If the income is less than or equal to 300000,
# it prints "Eligible". If the income is greater than 300000, it prints "Return Conditionally Eligible".
# If the age condition is false (i.e., age is less than 18), it directly prints "Not Eligible".
# There are no loops in this program; it simply evaluates the conditions once based on the user input.
```

Explanation

In this task, AI is used to explain how the conditions and loops work in the program. The explanation focuses only on the decision-making logic and loop behavior.

Task Description -3 (Documentation – File-Level Overview)

Prompt

Generate a complete Python script with multiple related functions that perform basic mathematical operations and include user input for testing the functions.

#Write a brief and clear file-level overview for the following Python script.

#Summarize the overall purpose and functionality.

Code

```python
#Generate a complete Python script with multiple related functions that perform basic mathematical operations and include user input for
#Function to add two numbers
def add(a, b):
    return a + b
#Function to subtract two numbers
def subtract(a, b):
    return a - b
#Function to multiply two numbers
def multiply(a, b):
    return a * b
#Function to divide two numbers
def divide(a, b):
    if b != 0:
        return a / b
    else:
        return "Cannot divide by zero"
#User input
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
#Perform operations
print("Addition:", add(num1, num2))
print("Subtraction:", subtract(num1, num2))
print("Multiplication:", multiply(num1, num2))
print("Division:", divide(num1, num2))
```

Output

```
Enter first number: 22
Enter second number: 43
Addition: 65.0
Subtraction: -21.0
Multiplication: 946.0
Division: 0.5116279069767442
```

```python
#Write a brief and clear file-level overview for the following Python script.
#Summarize the overall purpose and functionality.

#This Python script defines four basic mathematical functions (add, subtract, multiply, divide) and allows the user to input two numbers.
#It then performs all four operations on the numbers and displays the results.
# The script demonstrates user interaction and basic arithmetic operations in Python.
```

Explanation

First, AI is used to generate a complete Python script without any documentation. Then, another prompt is used to create a file-level overview describing the script's purpose.The generated overview is placed at the top of the file.
This helps improve readability and documentation quality.

Task Description -4 (Documentation – Refine Existing Documentation)

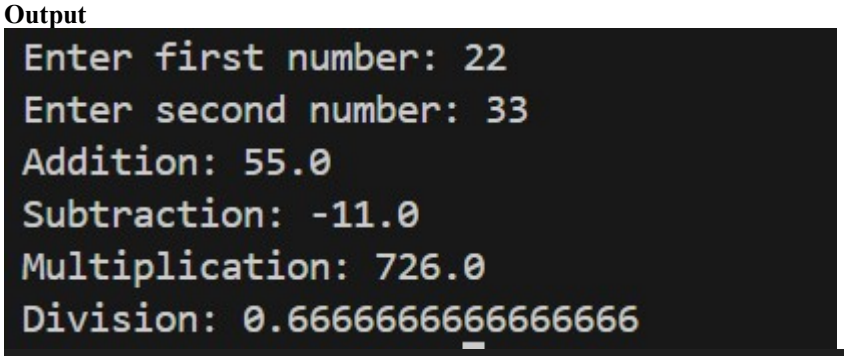
Prompt

Generate a Python script with 3–4 functions and include basic, short, or unclear comments.Rewrite the documentation and comments in the following Python code to improve clarity and consistency. Make the comments clear, concise, and professionally written.

Code

```python
#Generate a Python script with 3-4 functions and include basic, short, or unclear comments.
#Function to add two numbers
def add(a, b):
    return a + b
#Function to subtract two numbers
def subtract(a, b):
    return a - b
#Function to multiply two numbers
def multiply(a, b):
    return a * b
#Function to divide two numbers
def divide(a, b):
    if b != 0:
        return a / b
    else:
        return "Cannot divide by zero"
#User input
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
#Perform operations
print("Addition:", add(num1, num2))
print("Subtraction:", subtract(num1, num2))
print("Multiplication:", multiply(num1, num2))
print("Division:", divide(num1, num2))
```

Output




```python
#Rewrite the documentation and comments in the following Python code to improve clarity and consistency.
#Make the comments clear, concise, and professionally written.
#Function to add two numbers
def add(a, b):
    """Returns the sum of a and b."""
    return a + b
#Function to subtract two numbers\
def subtract(a, b):
    """Returns the difference of a and b."""
    return a - b
#Function to multiply two numbers
def multiply(a, b):
    """Returns the product of a and b."""
    return a * b
#Function to divide two numbers
def divide(a, b):
    """Returns the quotient of a and b. If b is zero, returns an error message."""
    if b != 0:
        return a / b
    else:
        return "Cannot divide by zero"
#User input
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
#Perform operations and display results
print("Addition:", add(num1, num2))
print("Subtraction:", subtract(num1, num2))
print("Multiplication:", multiply(num1, num2))
print("Division:", divide(num1, num2))
```

Explanation

First, AI is used to generate Python code with basic or unclear comments.
Then, another prompt is used to improve the clarity and consistency of those comments.
The logic of the code remains the same, but the documentation becomes clearer and more professional.This helps make the code easier to understand and maintain.

Task Description -5 (Documentation – Prompt Detail Impact Study)

Prompt

Generate a simple Python function that performs a real-world calculationWrite a clear and complete docstring for the following Python function

Code

```
#Generate a simple Python function that performs a real-world calculation
#Function to calculate the area of a circle
import math
from ssl import Purpose
def calculate_circle_area(radius):
    """Returns the area of a circle given its radius."""
    if radius < 0:
        return "Radius cannot be negative"
    return math.pi * (radius ** 2)
#User input
radius = float(input("Enter the radius of the circle: "))
#Calculate and display the area
area = calculate_circle_area(radius)
print("The area of the circle is:", area)
```

Output

```
#Write a clear and complete docstring for the following Python function.
def calculate_circle_area(radius):
    """Purpose:
    Calculates the area of a circle based on the given radius.

    Parameters:
    radius (float): The radius of the circle. Must be a non-negative number.

    Returns:
    float: The area of the circle calculated using the formula A = π * r^2.
    str: An error message if the radius is negative.

    Exceptions:
    ValueError: Raised if the input radius is negative.
    """
    if radius < 0:
        return "Radius cannot be negative"
    return math.pi * (radius ** 2)
```

Explanation

First, a Python function is generated using AI without any documentation.
Then, two different prompt are used to document the same function.
The outputs are compared to observe how prompt detail affects documentation quality.
This shows that more detailed prompts produce clearer and more complete documentation.