

ASSIGNMENT 9.3

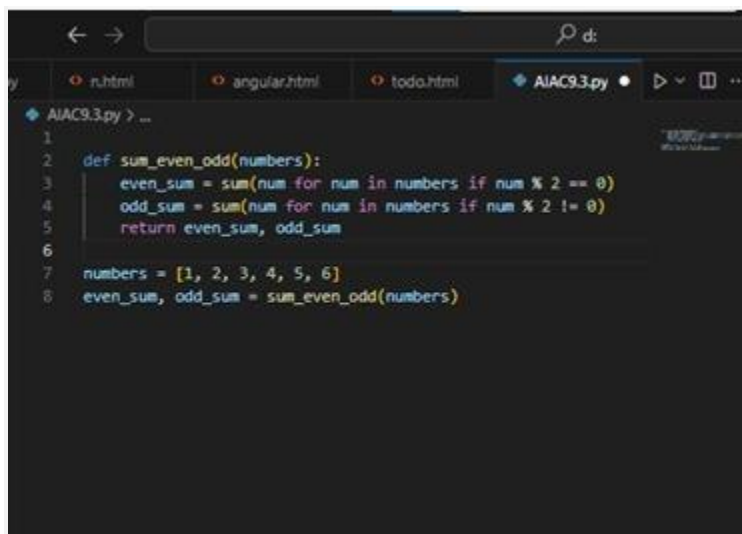
K.CHANDRIKAA

2403A51227

TASK 1:

Task Description#1 Basic Docstring Generation

- Write python function to return sum of even and odd numbers in the given list.
- Incorporate manual docstring in code with Google Style
- Use an AI-assisted tool (e.g., Copilot, Cursor AI) to generate a docstring describing the function.
- Compare the AI-generated docstring with your manually written one.

A screenshot of a code editor window with a dark theme. The editor shows a Python file named 'AIAC9.3.py'. The code defines a function 'sum_even_odd' with a docstring in Google style. The docstring describes the function's purpose, parameters, and return values. The function uses list comprehensions to calculate the sum of even and odd numbers. Below the function, a list 'numbers' is defined with values [1, 2, 3, 4, 5, 6], and the function is called to calculate the sums.

```
1 def sum_even_odd(numbers):  
2     """  
3     Calculate the sum of even and odd numbers in a list.  
4     Parameters:  
5     - numbers: A list of integers.  
6     Returns:  
7     - even_sum: The sum of even numbers.  
8     - odd_sum: The sum of odd numbers.  
9     """  
10    even_sum = sum(num for num in numbers if num % 2 == 0)  
11    odd_sum = sum(num for num in numbers if num % 2 != 0)  
12    return even_sum, odd_sum  
13  
14    numbers = [1, 2, 3, 4, 5, 6]  
15    even_sum, odd_sum = sum_even_odd(numbers)
```

Docstrings:

1. Manual docstrings:

```
def sum_even_odd(numbers): # Function to sum even and odd numbers separately
    even_sum = sum(num for num in numbers if num % 2 == 0) #Logic to sum even numbers
    odd_sum = sum(num for num in numbers if num % 2 != 0) #Logic to sum odd numbers
    return even_sum, odd_sum

numbers = [1, 2, 3, 4, 5, 6] # Example list of numbers
even_sum, odd_sum = sum_even_odd(numbers) # Call the function and unpack the results
print(f"Sum of even numbers: {even_sum}") # Print the sum of even numbers
print(f"Sum of odd numbers: {odd_sum}") # Print the sum of odd numbers
```

2. AI GENERATED DOCSTRINGS:

The screenshot shows a code editor with a file explorer at the top containing files like AIAC 8.2.py, n.html, angular.html, todo.html, AIAC9.3.py (selected), and Release Notes: 1.104.0. The main editor area displays the file AIAC9.3.py with the following content:

```
1
2 def sum_even_odd(numbers):
3     """
4     Calculates the sum of even and odd numbers in a given iterable.
5
6     Args:
7         numbers (iterable of int): A sequence of integers to be processed.
8
9     Returns:
10        tuple: A tuple containing two integers:
11            - The sum of even numbers.
12            - The sum of odd numbers.
13
14    Example:
15        >>> sum_even_odd([1, 2, 3, 4, 5])
16        (6, 9)
17    """
18    even_sum = sum(num for num in numbers if num % 2 == 0)
19    odd_sum = sum(num for num in numbers if num % 2 != 0)
20    return even_sum, odd_sum
21
22 numbers = [1, 2, 3, 4, 5, 6]
23 even_sum, odd_sum = sum_even_odd(numbers)
```

TASK 2:

Task Description#2 Automatic Inline Comments

- Write python program for sru_student class with attributes like name, roll no., hostel_status and fee_update method and display_details method.
- Write comments manually for each line/code block
- Ask an AI tool to add inline comments explaining each line/step.
- Compare the AI-generated comments with your manually written one

DOCSTRINGS:

1. MANUAL DOCSTRING

```

AIAC9.3.py > ...
1  class SRUStudent: #creating a class
2      def __init__(self, name, roll_no, hostel_status): #using the init constructor
3          self.name = name #defining attributes of the class using self.
4          self.roll_no = roll_no
5          self.hostel_status = hostel_status
6          self.fee_status = "Unpaid"
7
8      def fee_update(self, status): #defining a method
9          self.fee_status = status
10
11     def display_details(self): #defining another method
12         print(f"Name: {self.name}")
13         print(f"Roll No: {self.roll_no}")
14         print(f"Hostel Status: {self.hostel_status}")
15         print(f"Fee Status: {self.fee_status}")
16
17     student1 = SRUStudent("Alice", 101, "In-House") #creating an object.
18     student1.display_details()
19     student1.fee_update("Paid")
20     student1.display_details()
21

```

The above is manual docstring

2. AI GENERATED DOCSTRING:

```

AIAC 8.2.py  n.html  angular.html  todo.html  AIAC9.3.py  Release Notes: 1.104.0
AIAC9.3.py > SRUStudent > _init_
1  class SRUStudent:
2      """
3      SRUStudent represents a student at SRU with attributes for name, roll number, hostel status, and fee status.
4      Attributes:
5          name (str): The name of the student.
6          roll_no (str): The roll number assigned to the student.
7          hostel_status (str): Indicates whether the student is a hostel resident.
8          fee_status (str): The current fee payment status ("Unpaid" by default).
9      Methods:
10         fee_update(status):
11             Updates the fee_status attribute to the given status.
12         display_details():
13             Prints the student's details including name, roll number, hostel status, and fee status.
14         """
15     def __init__(self, name, roll_no, hostel_status):
16         self.name = name
17         self.roll_no = roll_no
18         self.hostel_status = hostel_status
19         self.fee_status = "Unpaid"
20
21     def fee_update(self, status):
22         self.fee_status = status
23
24     def display_details(self):
25         print(f"Name: {self.name}")
26         print(f"Roll No: {self.roll_no}")
27         print(f"Hostel Status: {self.hostel_status}")
28         print(f"Fee Status: {self.fee_status}")
29
30     student1 = SRUStudent("Alice", 101, "In-House")
31     student1.display_details()
32     student1.fee_update("Paid")
33     student1.display_details()
34

```

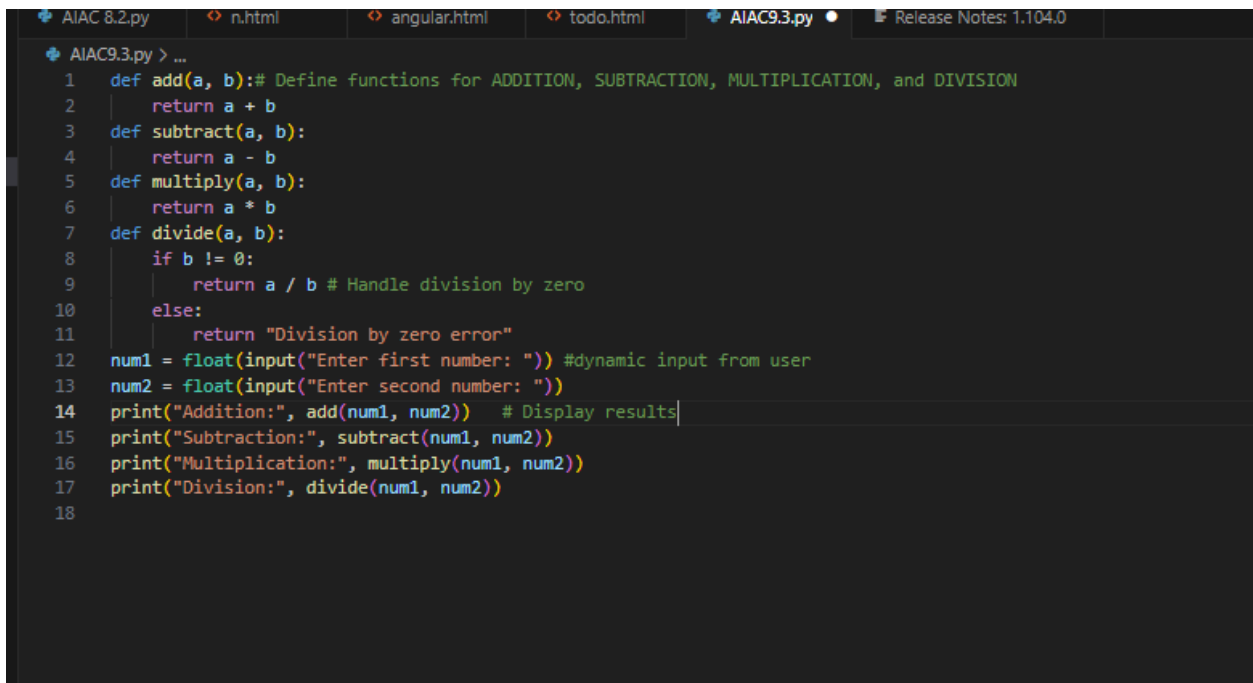
This is the AI GENERATED DOCSTRING.

TASK 3:

Task Description#3

- Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide).
- Incorporate manual docstring in code with NumPy Style
- Use AI assistance to generate a module-level docstring + individual function docstrings.
- Compare the AI-generated docstring with your manually written one

Manual docstrings:



```
AIAC 8.2.py  n.html  angular.html  todo.html  AIAC9.3.py  Release Notes: 1.104.0
AIAC9.3.py > ...
1  def add(a, b):# Define functions for ADDITION, SUBTRACTION, MULTIPLICATION, and DIVISION
2      return a + b
3  def subtract(a, b):
4      return a - b
5  def multiply(a, b):
6      return a * b
7  def divide(a, b):
8      if b != 0:
9          return a / b # Handle division by zero
10     else:
11         return "Division by zero error"
12  num1 = float(input("Enter first number: ")) #dynamic input from user
13  num2 = float(input("Enter second number: "))
14  print("Addition:", add(num1, num2))  # Display results
15  print("Subtraction:", subtract(num1, num2))
16  print("Multiplication:", multiply(num1, num2))
17  print("Division:", divide(num1, num2))
18
```

AI GENERATED DOCSTINGS:

```
1 def add(a, b):
2     '''This module provides basic arithmetic operations: addition, subtraction, multiplication, and division.
3     It prompts the user to input two numbers and displays the results of each operation.
4     Functions:
5         add(a, b): Returns the sum of a and b.
6         subtract(a, b): Returns the difference between a and b.
7         multiply(a, b): Returns the product of a and b.
8         divide(a, b): Returns the quotient of a divided by b, or an error message if division by zero occurs.
9     Usage:
10         The user is prompted to enter two numbers. The module then prints the results of addition, subtraction,
11         multiplication, and division using the provided functions.'''
12     return a + b
13 def subtract(a, b):
14     return a - b
15 def multiply(a, b):
16     return a * b
17 def divide(a, b):
18     """
19     Divides two numbers and returns the result.
20
21     Parameters:
22         a (float or int): The numerator.
23         b (float or int): The denominator.
24
25     Returns:
26         float: The result of division if b is not zero.
27         str: An error message if b is zero.
28     """
29     if b != 0:
30         return a / b
31     else:
32         return "Division by zero error"
33 num1 = float(input("Enter first number: "))
34 num2 = float(input("Enter second number: "))
35 print("Addition: ", add(num1, num2))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS