AI ASSISTED CODING

Assignment : 9.3 - Documentation Generation: Automatic documentation and code comments

YESWANTH KOYYA

2403A52098

24BTCAIAI05

TASK – 1 : Write python function to return sum of even and odd numbers in the given list without comments .

CODE, OUTPUT & DOC'STRING:

AI A-9.3.py > ...

```python
#task-1:Write python function to return sum of even and odd numbers in the given list.
def sum_even_odd(numbers):
    even_sum = 0
    odd_sum = 0

    for num in numbers:
        if num % 2 == 0:
            even_sum += num
        else:
            odd_sum += num

    return even_sum, odd_sum

even, odd = sum_even_odd([1, 2, 3, 4, 5, 6])
print(f"Even sum: {even}")
print(f"Odd sum: {odd}")
```

```
\"\"\"
Calculate the sum of even and odd numbers in a given list.

Args:
    numbers (list): A list of integers to process

Returns:
    tuple: A tuple containing (even_sum, odd_sum) where:
        - even_sum (int): Sum of all even numbers in the list
        - odd_sum (int): Sum of all odd numbers in the list

Example:
    >>> even, odd = sum_even_odd([1, 2, 3, 4, 5, 6])
    >>> print(f'Even sum: {even}')
    Even sum: 12
    >>> print(f'Odd sum: {odd}')
    Odd sum: 9
\"\"\"
```

```python
def sum_even_odd(numbers):
    even_sum = 0
    odd_sum = 0
    for number in numbers:
        if number % 2 == 0:
            even_sum += number
        else:
            odd_sum += number
    return even_sum, odd_sum

# Example usage:
my_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_sum, odd_sum = sum_even_odd(my_list)
print(f"Sum of even numbers: {even_sum}")
print(f"Sum of odd numbers: {odd_sum}")
```

```
Sum of even numbers: 30
Sum of odd numbers: 25
```

```python
def sum_even_odd(numbers):
    """
    Calculate the sum of even and odd numbers in a list.

    Args:
        numbers: A list of numbers.

    Returns:
        A tuple containing the sum of even numbers and the sum of odd numbers.
    """
    even_sum = 0  # Initialize sum for even numbers
    odd_sum = 0   # Initialize sum for odd numbers
    for number in numbers:
        if number % 2 == 0:
            even_sum += number  # Add even number to even_sum
        else:
            odd_sum += number   # Add odd number to odd_sum
    return even_sum, odd_sum  # Return both sums

# Example usage:
my_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_sum, odd_sum = sum_even_odd(my_list)
print(f"Sum of even numbers: {even_sum}")
print(f"Sum of odd numbers: {odd_sum}")
```

TASK – 2 : Write python program for sru_student class with attributes like name, roll no., hostel_status and fee_update method and display_details method with inline comments.

CODE DOC'STRING & OUTPUT :

```python
class sru_student:
    """A class to represent a student at SRU."""
    def __init__(self, name, roll_no, hostel_status):
        """Initializes a new sru_student object.
        Args:
            name (str): The name of the student.
            roll_no (str): The roll number of the student.
            hostel_status (bool): The hostel status of the student (True if in hostel, False otherwise)."""
        self.name = name  # Student's name
        self.roll_no = roll_no # Student's roll number
        self.hostel_status = hostel_status # Student's hostel status
        self.fees_paid = 0 # Initialize fees paid to 0
    def fee_update(self, amount):
        """Updates the fees paid by the student.
        Args:
            amount (float): The amount of fees paid."""
        self.fees_paid += amount # Add the paid amount to the total fees paid
        print(f"Fees updated for {self.name}. Total fees paid: {self.fees_paid}") # Print confirmation
    def display_details(self):
        """Displays the details of the student."""
        print("Student Details:") # Header for details
        print(f"Name: {self.name}") # Display student's name
        print(f"Roll Number: {self.roll_no}") # Display student's roll number
        print(f"Hostel Status: {'In Hostel' if self.hostel_status else 'Not in Hostel'}") # Display hostel status
        print(f"Fees Paid: {self.fees_paid}") # Display fees paid
# Example usage:
student1 = sru_student("Alice", "SRU123", True)
student1.display_details()
student1.fee_update(5000)
student1.display_details()
student2 = sru_student("Bob", "SRU456", False)
student2.display_details()
student2.fee_update(3000)
student2.display_details()
```

```
Student Details:
Name: Alice
Roll Number: SRU123
Hostel Status: In Hostel
Fees Paid: 0
Fees updated for Alice. Total fees paid: 5000
Student Details:
Name: Alice
Roll Number: SRU123
Hostel Status: In Hostel
Fees Paid: 5000
Student Details:
Name: Bob
Roll Number: SRU456
Hostel Status: Not in Hostel
Fees Paid: 0
Fees updated for Bob. Total fees paid: 3000
Student Details:
Name: Bob
Roll Number: SRU456
Hostel Status: Not in Hostel
Fees Paid: 3000
```

```python
class sru_student:
    def __init__(self, name, roll_no, hostel_status):
        self.name = name  # Student name
        self.roll_no = roll_no  # Student roll number
        self.hostel_status = hostel_status  # Whether student stays in hostel or not
        self.fee_amount = 0  # Initialize fee amount to 0
    def fee_update(self, amount):
        self.fee_amount = amount  # Update the fee amount
    def display_details(self):
        print(f"Name: {self.name}")  # Display student name
        print(f"Roll No: {self.roll_no}")  # Display roll number
        print(f"Hostel Status: {self.hostel_status}")  # Display hostel status
        print(f"Fee Amount: {self.fee_amount}")  # Display current fee amount
# Example usage
student1 = sru_student("John Doe", "SRU001", "Yes")  # Create student object
student1.fee_update(50000)  # Update fee amount
student1.display_details()  # Display all details
print("\n")  # Add blank line for separation
student2 = sru_student("Jane Smith", "SRU002", "No")  # Create another student
student2.fee_update(45000)  # Update fee amount
student2.display_details()  # Display all details
```

```
ers/HP/Documents/2403a52107/AI A-9.3.py"
Name: John Doe
Roll No: SRU001
Hostel Status: Yes
Fee Amount: 50000


Name: Jane Smith
Roll No: SRU002
Hostel Status: No
Fee Amount: 45000
```

TASK – 3 : Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide) and generate a module-level docstring + individual function docstrings.

CODE, OUTPUT & DOC'STRING

```python
import numpy as np
def add(a, b):
    return np.add(a, b)
def subtract(a, b):
    return np.subtract(a, b)
def multiply(a, b):
    return np.multiply(a, b)
def divide(a, b):
    if np.any(b == 0):
        print("Warning: Division by zero encountered.")
    return np.divide(a, b)
# Example usage:
num1 = 10
num2 = 5
print(f"{num1} + {num2} = {add(num1, num2)}")
print(f"{num1} - {num2} = {subtract(num1, num2)}")
print(f"{num1} * {num2} = {multiply(num1, num2)}")
print(f"{num1} / {num2} = {divide(num1, num2)}")
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])
print(f"{arr1} + {arr2} = {add(arr1, arr2)}")
print(f"{arr1} - {arr2} = {subtract(arr1, arr2)}")
print(f"{arr1} * {arr2} = {multiply(arr1, arr2)}")
print(f"{arr1} / {arr2} = {divide(arr1, arr2)}")
```

```
10 + 5 = 15
10 - 5 = 5
10 * 5 = 50
10 / 5 = 2.0
[1 2 3] + [4 5 6] = [5 7 9]
[1 2 3] - [4 5 6] = [-3 -3 -3]
[1 2 3] * [4 5 6] = [ 4 10 18]
[1 2 3] / [4 5 6] = [0.25 0.4  0.5 ]
```

:

```python
"""This module provides basic calculator functions using NumPy.
It includes functions for addition, subtraction, multiplication, and division."""
import numpy as np
def add(a, b):
    """Adds two numbers or NumPy arrays.
    Args:
        a: The first number or NumPy array.
        b: The second number or NumPy array.
    Returns:
        The sum of a and b."""
    return np.add(a, b)
def subtract(a, b):
    """Subtracts the second number or NumPy array from the first.
    Args:
        a: The first number or NumPy array.
        b: The second number or NumPy array.
    Returns:
        The difference between a and b."""
    return np.subtract(a, b)
def multiply(a, b):
    """Multiplies two numbers or NumPy arrays.
    Args:
        a: The first number or NumPy array.
        b: The second number or NumPy array.
    Returns:
        The product of a and b."""
    return np.multiply(a, b)
def divide(a, b):
    """Divides the first number or NumPy array by the second.
    Args:
        a: The first number or NumPy array (dividend).
        b: The second number or NumPy array (divisor).
    Returns:
        The quotient of a and b."""
    # Handle division by zero
    if np.any(b == 0):
        print("Warning: Division by zero encountered.")
    return np.divide(a, b)
# Example usage:
num1 = 10
num2 = 5
print(f"{num1} + {num2} = {add(num1, num2)}")
print(f"{num1} - {num2} = {subtract(num1, num2)}")
print(f"{num1} * {num2} = {multiply(num1, num2)}")
print(f"{num1} / {num2} = {divide(num1, num2)}")
```

```python
import numpy as np # Import the numpy library
def add(a, b):
    # This function adds two numbers or numpy arrays
    return np.add(a, b)

def subtract(a, b):
    # This function subtracts the second number/array from the first
    return np.subtract(a, b)

def multiply(a, b):
    # This function multiplies two numbers or numpy arrays
    return np.multiply(a, b)
def divide(a, b):
    # This function divides the first number/array by the second
    # Handle division by zero
    if np.any(b == 0):
        print("Warning: Division by zero encountered.")
    return np.divide(a, b)
# Example usage:
num1 = 10 # First number
num2 = 5 # Second number
print(f"{num1} + {num2} = {add(num1, num2)}") # Print the sum
print(f"{num1} - {num2} = {subtract(num1, num2)}") # Print the difference
print(f"{num1} * {num2} = {multiply(num1, num2)}") # Print the product
print(f"{num1} / {num2} = {divide(num1, num2)}") # Print the quotient
arr1 = np.array([1, 2, 3]) # First array
arr2 = np.array([4, 5, 6]) # Second array
print(f"{arr1} + {arr2} = {add(arr1, arr2)}") # Print the sum of arrays
print(f"{arr1} - {arr2} = {subtract(arr1, arr2)}") # Print the difference of arrays
print(f"{arr1} * {arr2} = {multiply(arr1, arr2)}") # Print the product of arrays
print(f"{arr1} / {arr2} = {divide(arr1, arr2)}") # Print the quotient of arrays
```

```python
def add(a, b):
    return a + b

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

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

def divide(a, b):
    if b != 0:
        return a / b
    else:
        return "Error: Division by zero"

print("Calculator Functions:")
print(f"Add: 10 + 5 = {add(10, 5)}")
print(f"Subtract: 10 - 5 = {subtract(10, 5)}")
print(f"Multiply: 10 * 5 = {multiply(10, 5)}")
print(f"Divide: 10 / 5 = {divide(10, 5)}")
print(f"Divide by zero: 10 / 0 = {divide(10, 0)}")
```

```
Calculator Functions:
Add: 10 + 5 = 15
Subtract: 10 - 5 = 5
Multiply: 10 * 5 = 50
Divide: 10 / 5 = 2.0
Divide by zero: 10 / 0 = Error: Division by zero
```

```python
def add(a, b):
    """Add two numbers.
    Args:
        a (int/float): First number
        b (int/float): Second number
    Returns:
        int/float: Sum of a and b"""
    return a + b
def subtract(a, b):
    """Subtract second number from first number.
    Args:
        a (int/float): First number
        b (int/float): Second number
    Returns:
        int/float: Difference of a and b"""
    return a - b
def multiply(a, b):
    """Multiply two numbers.
    Args:
        a (int/float): First number
        b (int/float): Second number
    Returns:
        int/float: Product of a and b"""
    return a * b
def divide(a, b):
    """Divide first number by second number.
    Args:
        a (int/float): Dividend
        b (int/float): Divisor
    Returns:
        int/float or str: Quotient of a and b, or error message if division by zero"""
    if b != 0:
        return a / b
    else:
        return "Error: Division by zero"
print("Calculator Functions:")
print(f"Add: 10 + 5 = {add(10, 5)}")
print(f"Subtract: 10 - 5 = {subtract(10, 5)}")
print(f"Multiply: 10 * 5 = {multiply(10, 5)}")
print(f"Divide: 10 / 5 = {divide(10, 5)}")
print(f"Divide by zero: 10 / 0 = {divide(10, 0)}")
```