Assignment-9

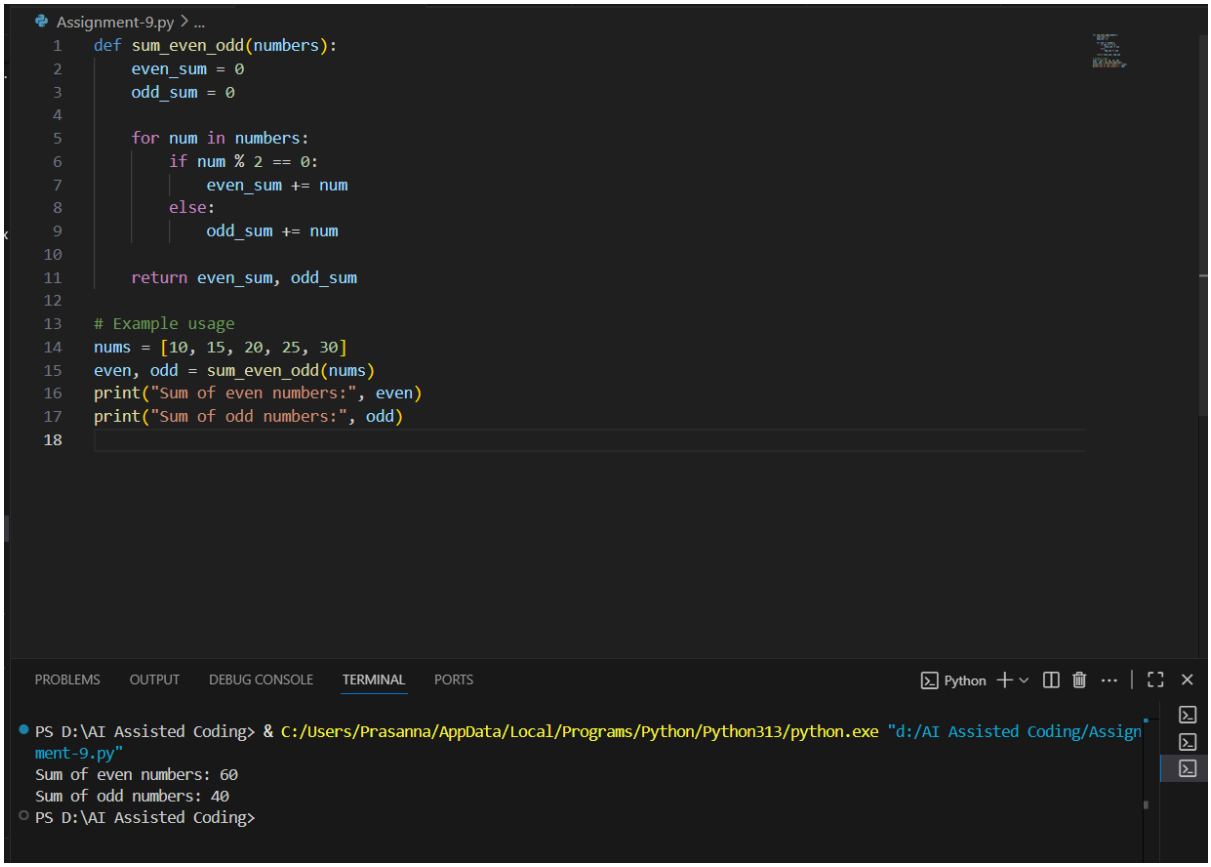
Task-1:

Prompt:

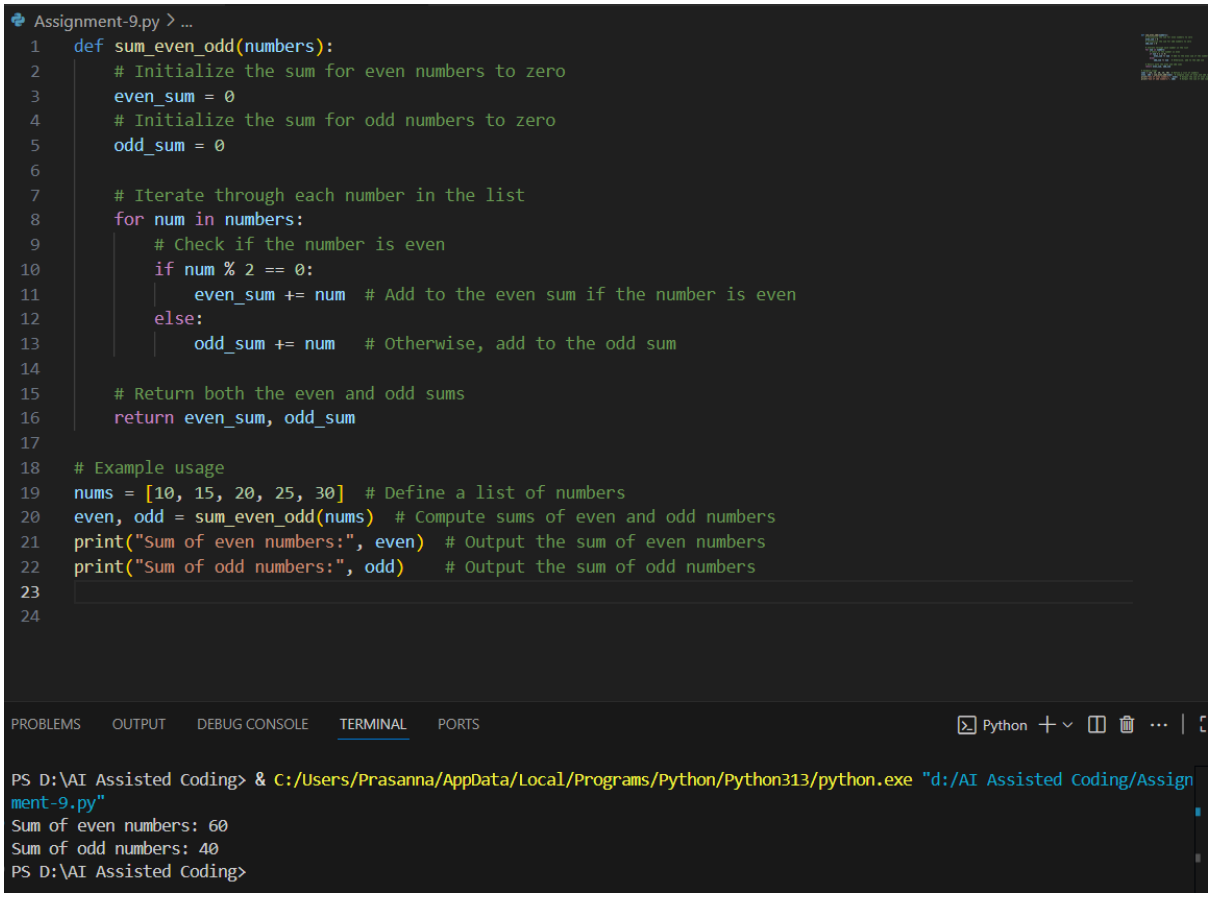
Write python function to return sum of even and odd numbers in the given list and compare the code without comments and with inline comments

Code and Output: **GitHub Copilot code**:

->Without Comments



->With Comments



Assignment-9.py > ...

```python
def sum_even_odd(numbers):
    # Initialize the sum for even numbers to zero
    even_sum = 0
    # Initialize the sum for odd numbers to zero
    odd_sum = 0

    # Iterate through each number in the list
    for num in numbers:
        # Check if the number is even
        if num % 2 == 0:
            even_sum += num  # Add to the even sum if the number is even
        else:
            odd_sum += num   # Otherwise, add to the odd sum

    # Return both the even and odd sums
    return even_sum, odd_sum

# Example usage
nums = [10, 15, 20, 25, 30]  # Define a list of numbers
even, odd = sum_even_odd(nums)  # Compute sums of even and odd numbers
print("Sum of even numbers:", even)  # Output the sum of even numbers
print("Sum of odd numbers:", odd)    # Output the sum of odd numbers

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS D:\AI Assisted Coding> & C:/Users/Prasanna/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Assisted Coding/Assign
ment-9.py"
Sum of even numbers: 60
Sum of odd numbers: 40
PS D:\AI Assisted Coding>
```

Google Colab Code:

->Without comments

```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 of the function
numbers_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
even_sum, odd_sum = sum_even_odd(numbers_list)

print(f"Even sum: {even_sum}")
print(f"Odd sum: {odd_sum}")
```

```
Even sum: 30
Odd sum: 25
```

->With comments

```python
# Define a function called sum_even_odd that takes a list of numbers as input
def sum_even_odd(numbers):
    # Initialize variables to store the sum of even and odd numbers
    even_sum = 0
    odd_sum = 0
    # Iterate through each number in the input list
    for number in numbers:
        # Check if the number is even
        if number % 2 == 0:
            # If even, add it to the even_sum
            even_sum += number
        else:
            # If odd, add it to the odd_sum
            odd_sum += number
    # Return the calculated even_sum and odd_sum
    return even_sum, odd_sum

# Example usage of the function
# Create a list of numbers
numbers_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
# Call the sum_even_odd function with the list and store the results
even_sum, odd_sum = sum_even_odd(numbers_list)

# Print the calculated even sum
print(f"Even sum: {even_sum}")
# Print the calculated odd sum
print(f"Odd sum: {odd_sum}")
```

```
Even sum: 30
Odd sum: 25
```

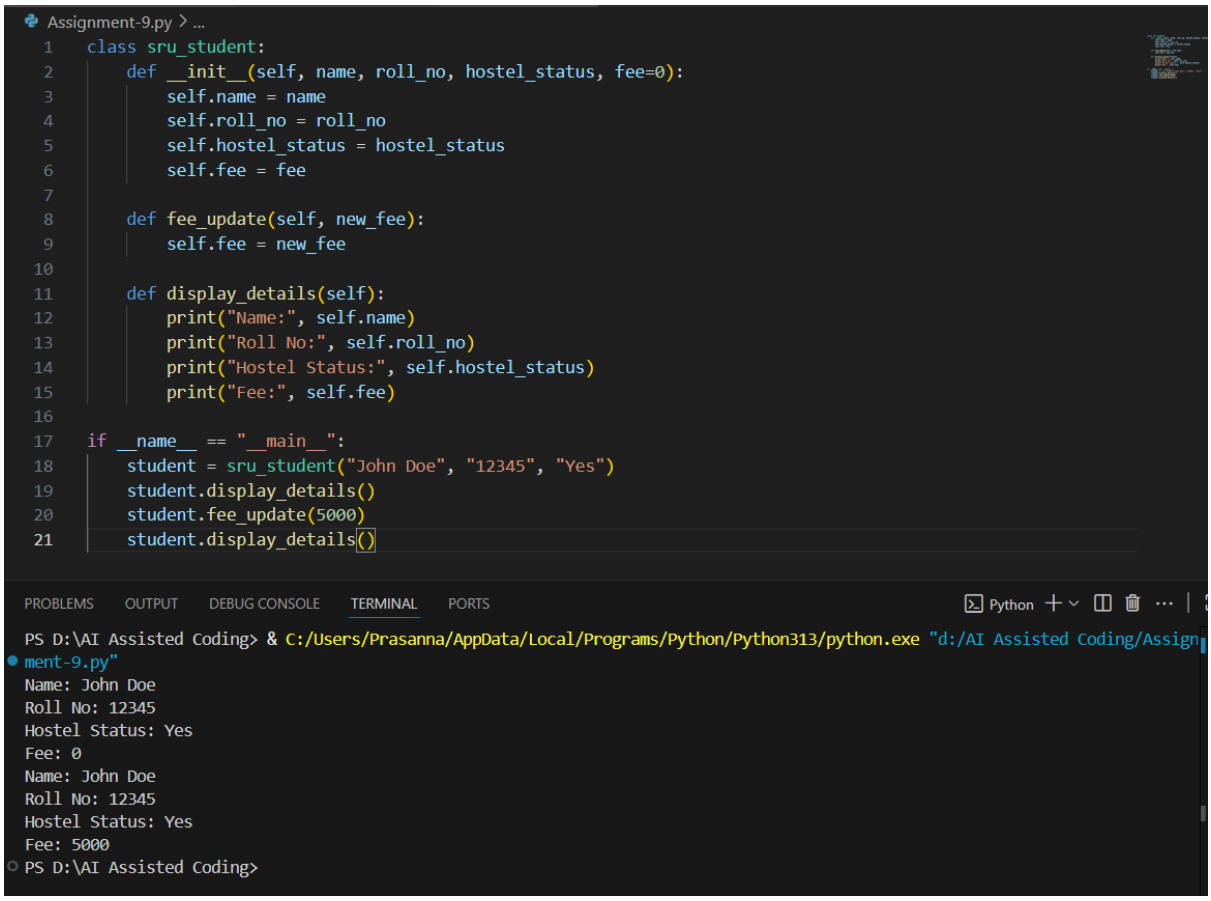
Task 2:

- Prompt:
- Write python program for **sru_student** class with attributes like name, roll no., hostel_status and **fee_update** method and **display_details** method and compare the code without comments and with inline comments

- Code and Output:

GitHub Copilot Code

->Without Comments



->With comments

Assignment-9.py > ...

```python
# This class represents an SRU student with details like name, roll number, hostel status, and fee.
class sru_student:
    # The constructor initializes the student's attributes.
    def __init__(self, name, roll_no, hostel_status, fee=0):
        self.name = name                  # Student's name
        self.roll_no = roll_no            # Student's roll number
        self.hostel_status = hostel_status  # Indicates if the student is in the hostel
        self.fee = fee                    # Student's fee, default is 0

    # Method to update the fee of the student.
    def fee_update(self, new_fee):
        self.fee = new_fee

    # Method to display the student's details.
    def display_details(self):
        print("Name:", self.name)
        print("Roll No:", self.roll_no)
        print("Hostel Status:", self.hostel_status)
        print("Fee:", self.fee)

# The main block to run the program.
if __name__ == "__main__":
    # Creating an instance of sru_student with initial details.
    student = sru_student("John Doe", "12345", "Yes")
    # Displaying the initial details of the student.
    student.display_details()
    # Updating the fee for the student.
    student.fee_update(5000)
    # Displaying the updated details of the student.
    student.display_details()
```

```
PS D:\AI Assisted Coding> & C:/Users/Prasanna/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Assisted Coding/Assignment-9.py"
Name: John Doe
Roll No: 12345
Hostel Status: Yes
Fee: 0
Name: John Doe
Roll No: 12345
Hostel Status: Yes
Fee: 5000
PS D:\AI Assisted Coding>
```

Google Colab Code:

->Without comments

```
class sru_student:
    def __init__(self, name, roll_no, hostel_status):
        self.name = name
        self.roll_no = roll_no
        self.hostel_status = hostel_status
        self.fee_status = "Unpaid"

    def fee_update(self, status):
        self.fee_status = status

    def display_details(self):
        print(f"Name: {self.name}")
        print(f"Roll No.: {self.roll_no}")
        print(f"Hostel Status: {self.hostel_status}")
        print(f"Fee Status: {self.fee_status}")
        # Create an instance of the sru_student class
student1 = sru_student("Alice", "SRU123", "Hosteller")

# Display the initial details of the student
print("Initial details:")
student1.display_details()

# Update the fee status of the student
student1.fee_update("Paid")

# Display the updated details of the student
print("\nUpdated details:")
student1.display_details()
```

```
Initial details:
Name: Alice
Roll No.: SRU123
Hostel Status: Hosteller
Fee Status: Unpaid

Updated details:
Name: Alice
Roll No.: SRU123
Hostel Status: Hosteller
Fee Status: Paid
```

->With comments

```python
# Define a class named sru_student
class sru_student:
    # The __init__ method is the constructor of the class.
    # It is called when a new object of the class is created.
    def __init__(self, name, roll_no, hostel_status):
        # Initialize the attributes of the student object
        self.name = name
        self.roll_no = roll_no
        self.hostel_status = hostel_status
        # Set the initial fee status to "Unpaid"
        self.fee_status = "Unpaid"

    # Define a method to update the fee status of the student
    def fee_update(self, status):
        # Update the fee_status attribute with the provided status
        self.fee_status = status

    # Define a method to display the details of the student
    def display_details(self):
        # Print the student's name
        print(f"Name: {self.name}")
        # Print the student's roll number
        print(f"Roll No.: {self.roll_no}")
        # Print the student's hostel status
        print(f"Hostel Status: {self.hostel_status}")
        # Print the student's fee status
        print(f"Fee Status: {self.fee_status}")

# Example usage of the sru_student class

# Create an instance (object) of the sru_student class with initial details
student1 = sru_student("Alice", "SRU123", "Hosteller")
```

```python
# Display the initial details of the student using the display_details method
print("Initial details:")
student1.display_details()

# Update the fee status of the student using the fee_update method
student1.fee_update("Paid")

# Display the updated details of the student
print("\nUpdated details:")
student1.display_details()
```

```
Initial details:
Name: Alice
Roll No.: SRU123
Hostel Status: Hosteller
Fee Status: Unpaid

Updated details:
Name: Alice
Roll No.: SRU123
Hostel Status: Hosteller
Fee Status: Paid
```

Task 3:

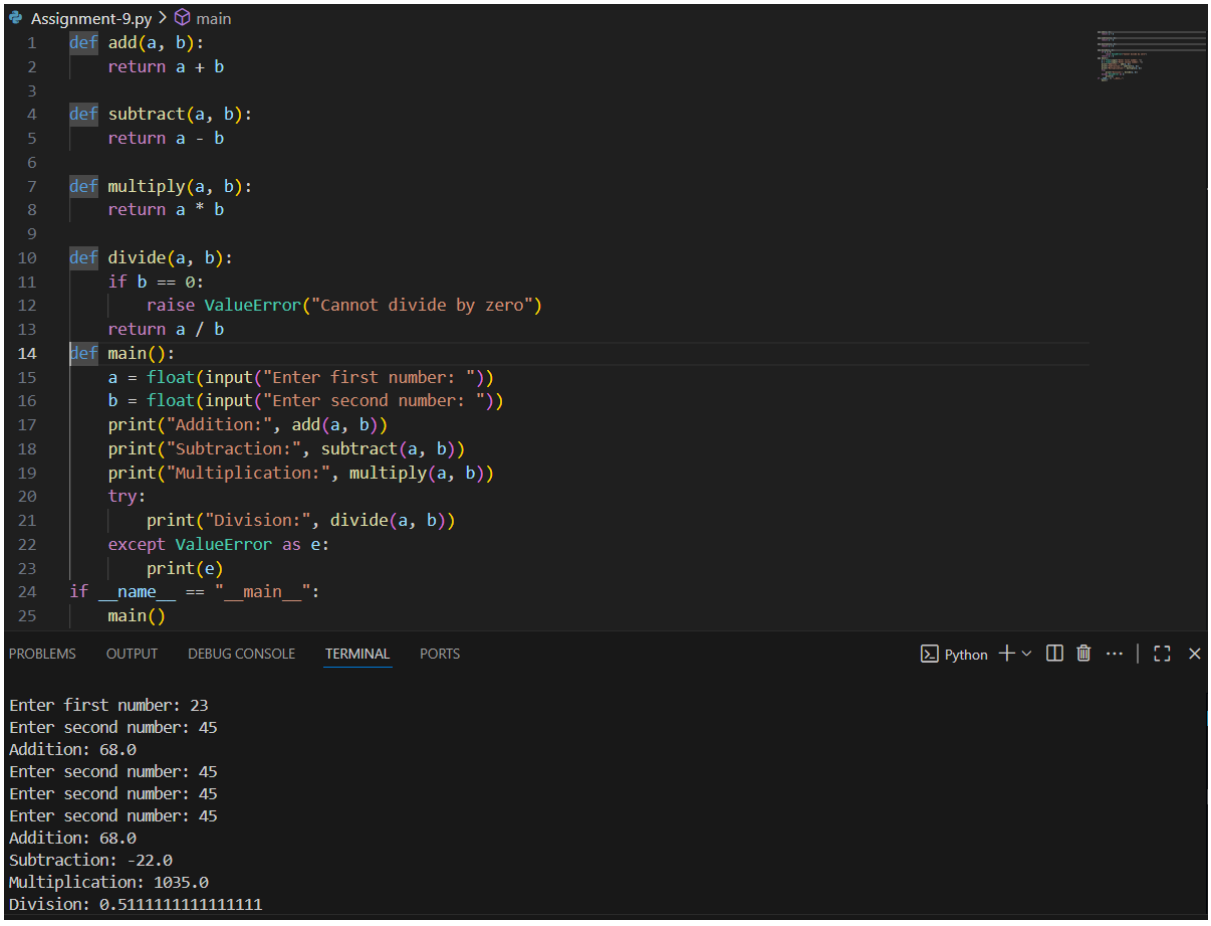
Prompt:

- Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide). And compare the code without comments and with inline comments

Code and Output:

GitHub Copilot code

->Without Comments



->With Comments:

Assignment-9.py > ...

```python
"""
Calculator script with basic arithmetic operations.
"""

def add(a, b):
    """Return the sum of a and b."""
    return a + b

def subtract(a, b):
    """Return the difference of a and b."""
    return a - b

def multiply(a, b):
    """Return the product of a and b."""
    return a * b

def divide(a, b):
    """Return the division of a by b. Raises an error if b is zero."""
    if b == 0:
        raise ValueError("Cannot divide by zero")
    return a / b

def main():
    """Main function that interacts with the user to perform calculations."""
    a = float(input("Enter first number: "))  # Get the first number from the user
    b = float(input("Enter second number: "))  # Get the second number from the user
    print("Addition:", add(a, b))               # Print the result of addition
    print("Subtraction:", subtract(a, b))       # Print the result of subtraction
    print("Multiplication:", multiply(a, b))    # Print the result of multiplication
    try:
        print("Division:", divide(a, b))        # Try dividing and print the result
    except ValueError as e:
        print(e)                                # Print the error if division by zero occurs

if __name__ == "__main__":
    main()  # Run the main function if this script is executed directly
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS D:\AI Assisted Coding> & C:/Users/Prasanna/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Assisted Coding/Assignment-9
.py"
Enter first number: 7
Enter second number: 6
Addition: 13.0
Subtraction: 1.0
Multiplication: 42.0
Division: 1.1666666666666667
PS D:\AI Assisted Coding>
```

Google Colab Code:

->Without Comments:

```python
def add(x, y):
    return x + y

def subtract(x, y):
    return x - y

def multiply(x, y):
    return x * y

def divide(x, y):
    if y == 0:
        return "Error: Division by zero"
    return x / y

num1 = 10
num2 = 5

print("Addition:", add(num1, num2))
print("Subtraction:", subtract(num1, num2))
print("Multiplication:", multiply(num1, num2))
print("Division:", divide(num1, num2))
print("Division by zero:", divide(num1, 0))
```

```
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0
Division by zero: Error: Division by zero
```

->With Comments:

```python
# Define a function for addition
def add(x, y):
    return x + y

# Define a function for subtraction
def subtract(x, y):
    return x - y

# Define a function for multiplication
def multiply(x, y):
    return x * y

# Define a function for division
def divide(x, y):
    # Check if the divisor is zero to avoid division by zero error
    if y == 0:
        return "Error: Division by zero"
    return x / y

# Example usage of the calculator functions
num1 = 10
num2 = 5

# Perform addition and print the result
print("Addition:", add(num1, num2))
# Perform subtraction and print the result
print("Subtraction:", subtract(num1, num2))
# Perform multiplication and print the result
print("Multiplication:", multiply(num1, num2))
# Perform division and print the result
print("Division:", divide(num1, num2))
```

```python
# Perform division and print the result
print("Division:", divide(num1, num2))
# Demonstrate division by zero and print the result
print("Division by zero:", divide(num1, 0))
```

```
Addition: 15
Subtraction: 5
Multiplication: 50
Division: 2.0
Division by zero: Error: Division by zero
```