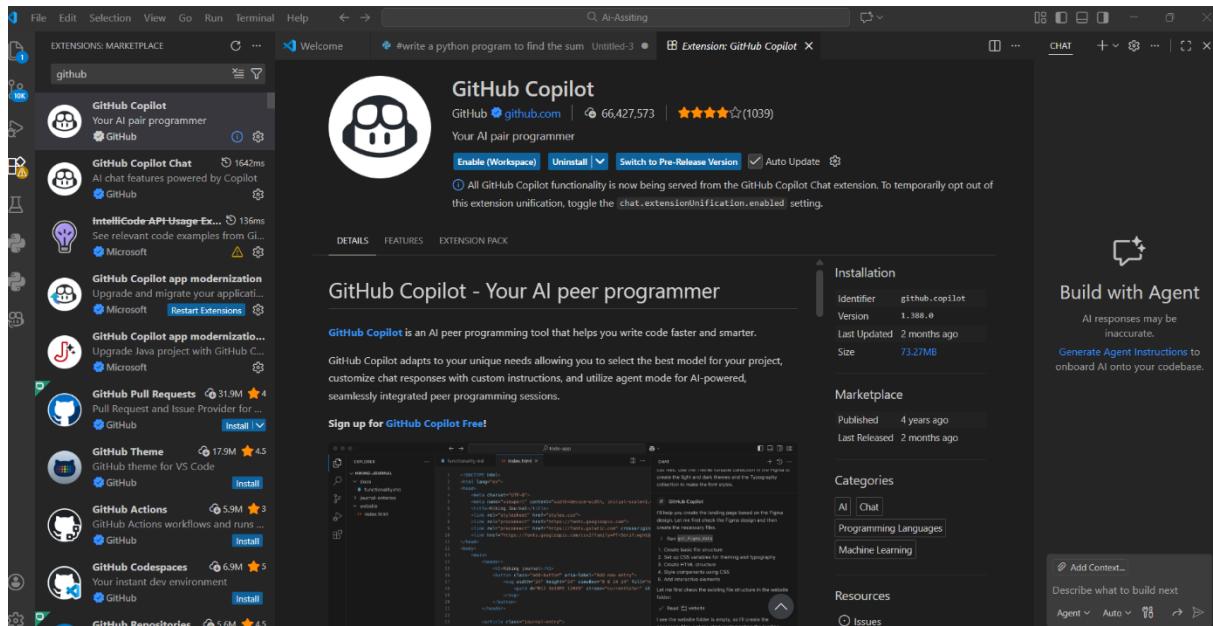


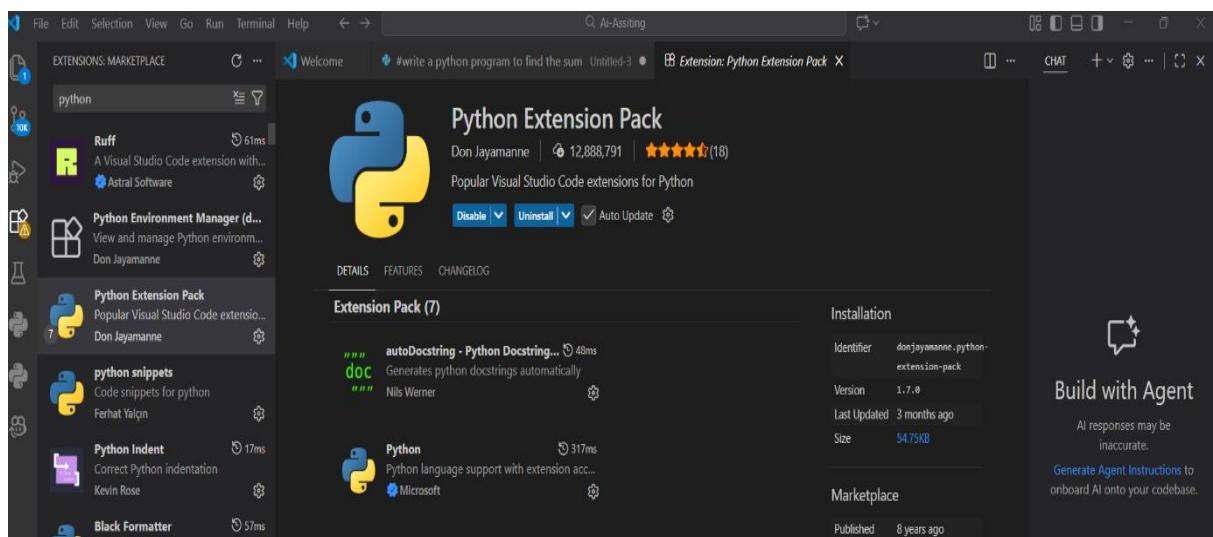
R Akshay

Lab Assignment - 1.3

- Downloaded the GitHub Copilot extension in vscode



- Downloaded the Python Extension pack in vscode



- Implementation of code with the extension pack

```

1 #write a python program to find the sum Untitled-3
2 def sum_of_two_numbers(num1, num2):
3     return num1 + num2

```

- Implementation of code using github copilot extension

```

1 #write a python script that takes a list of integers as input and returns a new list containing only
2 def filter_even_numbers(numbers):
3     return [num for num in numbers if num % 2 == 0]
4
5 # Example usage:
6 input_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
7 even_numbers = filter_even_numbers(input_list)
8 print("Even numbers:", even_numbers)

```

- You are asked to write a quick numerical sequence generator for a learning platform prototype.

Fibonacci series 5 steps

Task 1: AI-Generated Logic Without Modularization (Fibonacci Sequence Without Functions)

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Marketplace Sidebar:** Shows various GitHub extensions like GitHub Copilot, GitHub Pull Requests, GitHub Theme, etc.
- Code Editor:** A Python script titled "Untitled-1" containing the following code:

```
#write a fibonacci series without using functions
n = int(input("Enter the number of terms in the Fibonacci series: "))
a, b = 0, 1
if n <= 0:
    print("Please enter a positive integer.")
elif n == 1:
    print("Fibonacci series up to", n, ":")
    print(a)
else:
```
- Build with Agent Panel:** A sidebar with a message: "Build with Agent" and "AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase."
- Terminal:** A small terminal window at the bottom right.

#output for the code of fibonacci series with using functions

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Marketplace Sidebar:** Shows various GitHub extensions like GitHub Copilot, GitHub Pull Requests, GitHub Theme, etc.
- Code Editor:** A Python script titled "fibonaciseries.py" containing the following code:

```
# write a program to generate fibonacci series up to n terms n value by user input
def fibonacci_series(n):
    fib_series = []
    a, b = 0, 1
    for _ in range(n):
        fib_series.append(a)
        a, b = b, a + b
    return fib_series
if __name__ == "__main__":
    n = int(input("Enter the number of terms for Fibonacci series: "))
    series = fibonacci_series(n)
    print("Fibonacci series up to", n, "terms:", series)
```
- Terminal:** A terminal window at the bottom showing the execution of the script and its output.

```
tti/OneDrive/Desktop/AI-Assisting/fibonaciseries.py
PS C:\Users\mutili\OneDrive\Desktop\AI-Assisting> & c:\Users\mutili\AppData\Local\Programs\Python\Python313\python.exe c:/Users/mutili/OneDrive/Desktop/AI-Assisting/fibonaciseries.py
PS C:\Users\mutili\OneDrive\Desktop\AI-Assisting> & c:\Users\mutili\AppData\Local\Programs\Python\Python313\python.exe c:/Users/mutili/OneDrive/Desktop/AI-Assisting/fibonaciseries.py
PS C:\Users\mutili\OneDrive\Desktop\AI-Assisting> & c:\Users\mutili\AppData\Local\Programs\Python\Python313\python.exe c:/Users/mutili/OneDrive/Desktop/AI-Assisting/fibonaciseries.py
Enter the number of terms for Fibonacci series: 5
Fibonacci series up to 5 terms: [0, 1, 1, 2, 3]
PS C:\Users\mutili\OneDrive\Desktop\AI-Assisting>
```
- Build with Agent Panel:** A sidebar with a message: "Build with Agent" and "AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase."
- Terminal:** A small terminal window at the bottom right.

Task 2: AI Code Optimization & Cleanup (Improving Efficiency)

```

1 #write a fibonacci series code in brute force way
2 def fibonacci_series(n):
3     if n <= 0:
4         return []
5     elif n == 1:
6         return [0]
7     elif n == 2:
8         return [0, 1]
9
10    fib_series = [0, 1]
11    for i in range(2, n):
12        next_fib = fib_series[i-1] + fib_series[i-2]
13        fib_series.append(next_fib)
14    return fib_series
15
16 #write a code in optimized way to reduce time complexity
17 def optimized_fibonacci_series(n):
18    fib_series = []
19    a, b = 0, 1
20    for _ in range(n):
21        fib_series.append(a)
22        a, b = b, a + b
23    return fib_series

```

Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

Task 3: Modular Design Using AI Assistance (Fibonacci Using Functions)

```

fibonaci.py > ...
1 #write a python program fibonacci series using function
2 def fibonacci_series(n):
3     fib_series = []
4     a, b = 0, 1
5     for _ in range(n):
6         fib_series.append(a)
7         a, b = b, a + b
8     return fib_series
9
10 if __name__ == "__main__":
11     n = int(input("Enter the number of terms for Fibonacci series: "))
12     series = fibonacci_series(n)
13     print("Fibonacci series up to: ", n, "terms:", series)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

python + □ □ ... □
tti/OneDrive/Desktop/Ai-Assiting/fibonaciseries.py
PS C:\Users\mutili\OneDrive\Desktop\Ai-Assiting> & C:\Users\mutili\AppData\Local\Programs\Python\Python313\python.exe c:/users/mutili/OneDrive/Desktop/Ai-Assiting/fibonaciseries.py
tti/OneDrive/Desktop/Ai-Assiting/fibonaciseries.py
PS C:\Users\mutili\OneDrive\Desktop\Ai-Assiting> & C:\Users\mutili\AppData\Local\Programs\Python\Python313\python.exe c:/users/mutili/OneDrive/Desktop/Ai-Assiting/fibonaciseries.py
tti/OneDrive/Desktop/Ai-Assiting/fibonaciseries.py
Enter the number of terms for Fibonacci series: 5
Fibonacci series up to 5 terms: [0, 1, 1, 2, 3]
tti/OneDrive/Desktop/Ai-Assiting/fibonaci.py
PS C:\Users\mutili\OneDrive\Desktop\Ai-Assiting> & C:\Users\mutili\AppData\Local\Programs\Python\Python313\python.exe c:/users/mutili/OneDrive/Desktop/Ai-Assiting/fibonaci.py
Enter the number of terms for Fibonacci series: []

```

Task 4: Comparative Analysis – Procedural vs Modular Fibonacci Code

The screenshot shows the VS Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** Q AI-Assiting
- Explorer:** OPEN EDITORS 1 unsaved
 - Welcome
 - fibonaciseries.py
 - fibonacci.py
 - AI-ASSITING
 - fibonacci.py
 - fibonaciseries.py
 - prompt.py
- Editor:** fibonacci.py


```
#write a python program fibonaccii series with function
def fibonacci_series(n):
    fib_series = []
    a, b = 0, 1
    for _ in range(n):
        fib_series.append(a)
        a, b = b, a + b
    return fib_series

if __name__ == "__main__":
    n = int(input("Enter the number of terms for Fibonacci series: "))
    series = fibonacci_series(n)
    print("Fibonacci series up to", n, "terms:", series)

#write a python program fibonaccii series without function
n = int(input("Enter the number of terms for Fibonacci series: "))
a, b = 0, 1
fib_series = []
for _ in range(n):
    fib_series.append(a)
    a, b = b, a + b
print("Fibonacci series up to", n, "terms:", fib_series)
```
- Terminal:** python + v 🗑️ ...
- Output:** PS C:\Users\multi\OneDrive\Desktop\Ai-Assiting & C:\Users\multi\AppData\Local\Programs\Python\Python313\python.exe c:/users/multi/OneDrive/Desktop/Ai-Assiting/fibonaciseries.py
- Problems:** PROBLEMS
- Debug Console:** DEBUG CONSOLE
- Terminal:** TERMINAL
- Ports:** PORTS
- Bottom Status Bar:** Ln 20, Col 57 Spaces:4 UTF-8 CRLF {} Python 8 Python 3

Task 5: AI-Generated Iterative vs Recursive Fibonacci Approaches (Different Algorithmic Approaches for Fibonacci Series)

The screenshot shows the VS Code interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Search Bar:** Q AI-Assiting
- Explorer:** OPEN EDITORS 1 unsaved
 - Welcome
 - fibonaciseries.py
 - fibonacci.py
 - AI-ASSITING
 - fibonacci.py
 - fibonaciseries.py
 - prompt.py
- Editor:** fibonacci.py


```
#write a python program fibonacci series using recursion
def fibonacci(n):
    if n <= 0:
        return []
    elif n == 1:
        return [0]
    elif n == 2:
        return [0, 1]
    else:
        fib_series = fibonacci(n - 1)
        fib_series.append(fib_series[-1] + fib_series[-2])
    return fib_series

if __name__ == "__main__":
    n = int(input("Enter the number of terms for Fibonacci series: "))
    series = fibonacci(n)
    print("Fibonacci series up to", n, "terms:", series)
```
- Terminal:** python + v 🗑️ ... CHAT
- Output:** PS C:\Users\multi\OneDrive\Desktop\Ai-Assiting & C:\Users\multi\AppData\Local\Programs\Python\Python313\python.exe c:/users/multi/OneDrive/Desktop/Ai-Assiting/fibonaciseries.py
- Problems:** PROBLEMS
- Debug Console:** DEBUG CONSOLE
- Terminal:** TERMINAL
- Ports:** PORTS
- Bottom Status Bar:** Ln 20, Col 57 Spaces:4 UTF-8 CRLF {} Python 8 Python 3
- Right Panel:**
 - Build with Agent:** AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase.
 - Agent Panel:** fibonacci.py + Describe what to build next Agent Auto ⚡