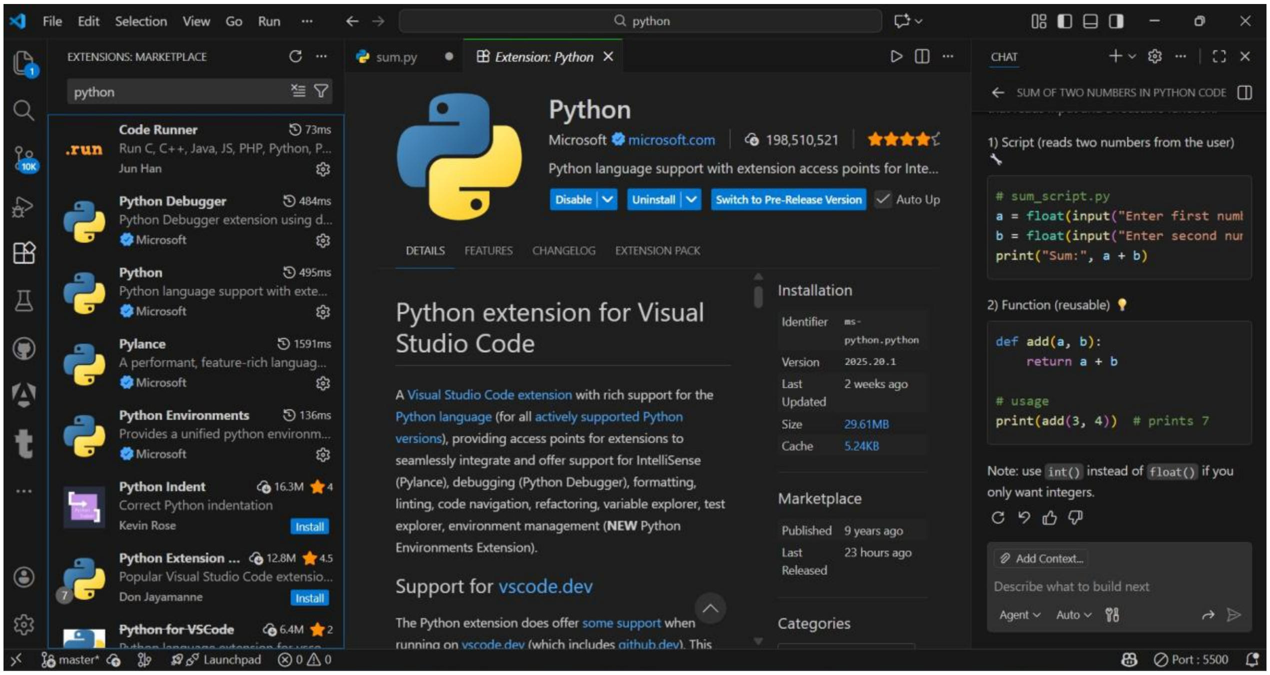
Lab Assignment 1.3

HT.NO :2303A51134

BATCH -27


File Edit Selection View Go Run ···
python
EXTENSIONS: MARKETPLACE
python
Code Runner 73ms
Run C, C++, Java, JS, PHP, Python, P...
Jun Han
Python Debugger 484ms
Python Debugger extension using d...
Microsoft
Python 495ms
Python language support with exte...
Microsoft
Pylance 1591ms
A performant, feature-rich languag...
Microsoft
Python Environments 136ms
Provides a unified python environm...
Microsoft
Python Indent 16.3M 4
Correct Python indentation
Kevin Rose
Install
Python Extension ... 12.8M 4.5
Popular Visual Studio Code extensio...
Don Jayamanne
Install
Python for VSCode 6.4M 2
sum.py
Extension: Python
Python
Microsoft microsoft.com 198,510,521
Python language support with extension access points for Inte...
Disable
Uninstall
Switch to Pre-Release Version
Auto Up
DETAILS FEATURES CHANGELOG EXTENSION PACK
CHAT
SUM OF TWO NUMBERS IN PYTHON CODE
Add Context...
Describe what to build next
Agent Auto
master* Launchpad 0 0
Port : 5500

Python extension for Visual Studio Code

A Visual Studio Code extension with rich support for the Python language (for all actively supported Python versions), providing access points for extensions to seamlessly integrate and offer support for IntelliSense (Pylance), debugging (Python Debugger), formatting, linting, code navigation, refactoring, variable explorer, test explorer, environment management (**NEW** Python Environments Extension).

Support for vscode.dev

The Python extension does offer some support when running on vscode.dev (which includes github.dev). This

Installation

Identifier	ms-python.python
Version	2025.20.1
Last Updated	2 weeks ago
Size	29.61MB
Cache	5.24KB

Marketplace

Published	9 years ago
Last Released	23 hours ago

Categories

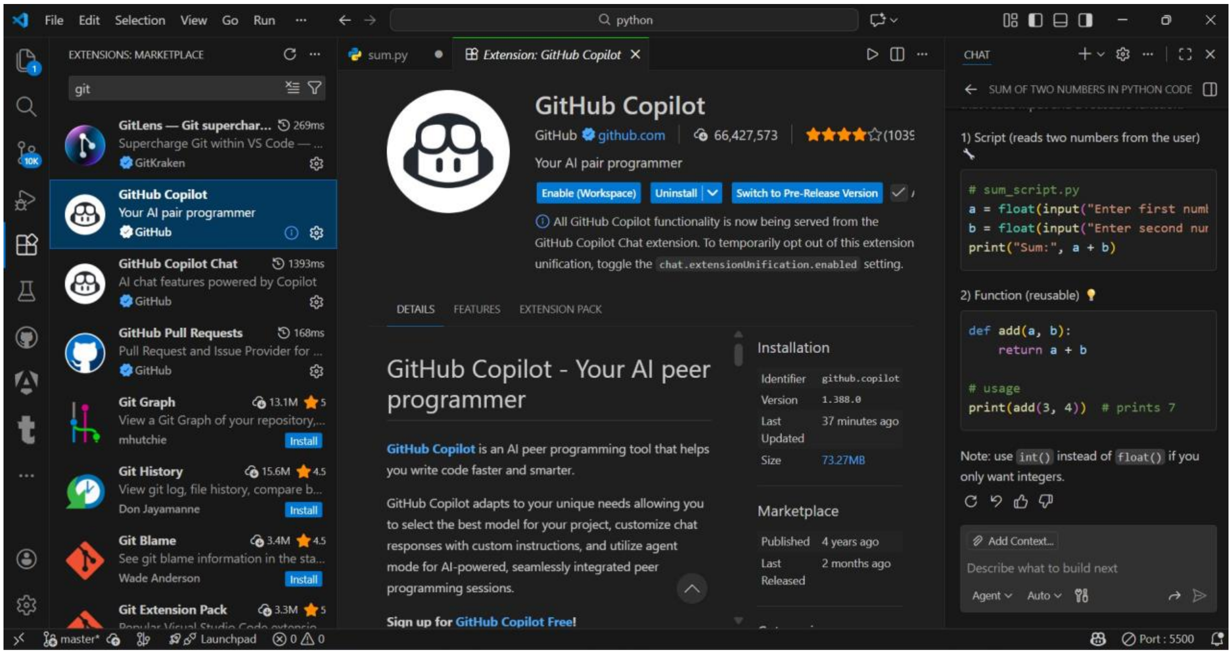
1) Script (reads two numbers from the user) 🔧

```
# sum_script.py
a = float(input("Enter first numl
b = float(input("Enter second nur
print("Sum:", a + b)
```

2) Function (reusable) 💡

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

# usage
print(add(3, 4))  # prints 7
```

Note: use `int()` instead of `float()` if you only want integers.


File Edit Selection View Go Run ···
python
EXTENSIONS: MARKETPLACE
git
GitLens — Git superchar... 269ms
Supercharge Git within VS Code — ...
GitKraken
GitHub Copilot
Your AI pair programmer
GitHub
GitHub Copilot Chat 1393ms
AI chat features powered by Copilot
GitHub
GitHub Pull Requests 168ms
Pull Request and Issue Provider for ...
GitHub
Git Graph 13.1M 5
View a Git Graph of your repository,...
mhutchie
Install
Git History 15.6M 4.5
View git log, file history, compare b...
Don Jayamanne
Install
Git Blame 3.4M 4.5
See git blame information in the sta...
Wade Anderson
Install
Git Extension Pack 3.3M 5
sum.py
Extension: GitHub Copilot
GitHub Copilot
GitHub github.com 66,427,573 (1039
Your AI pair programmer
Enable (Workspace)
Uninstall
Switch to Pre-Release Version
DETAILS FEATURES EXTENSION PACK
CHAT
SUM OF TWO NUMBERS IN PYTHON CODE
Add Context...
Describe what to build next
Agent Auto
master* Launchpad 0 0
Port : 5500

All GitHub Copilot functionality is now being served from the GitHub Copilot Chat extension. To temporarily opt out of this extension unification, toggle the `chat.extensionUnification.enabled` setting.

GitHub Copilot - Your AI peer programmer

GitHub Copilot is an AI peer programming tool that helps you write code faster and smarter.

GitHub Copilot adapts to your unique needs allowing you to select the best model for your project, customize chat responses with custom instructions, and utilize agent mode for AI-powered, seamlessly integrated peer programming sessions.

Sign up for GitHub Copilot Free!

Installation

Identifier	github.copilot
Version	1.388.0
Last Updated	37 minutes ago
Size	73.27MB

Marketplace

Published	4 years ago
Last Released	2 months ago

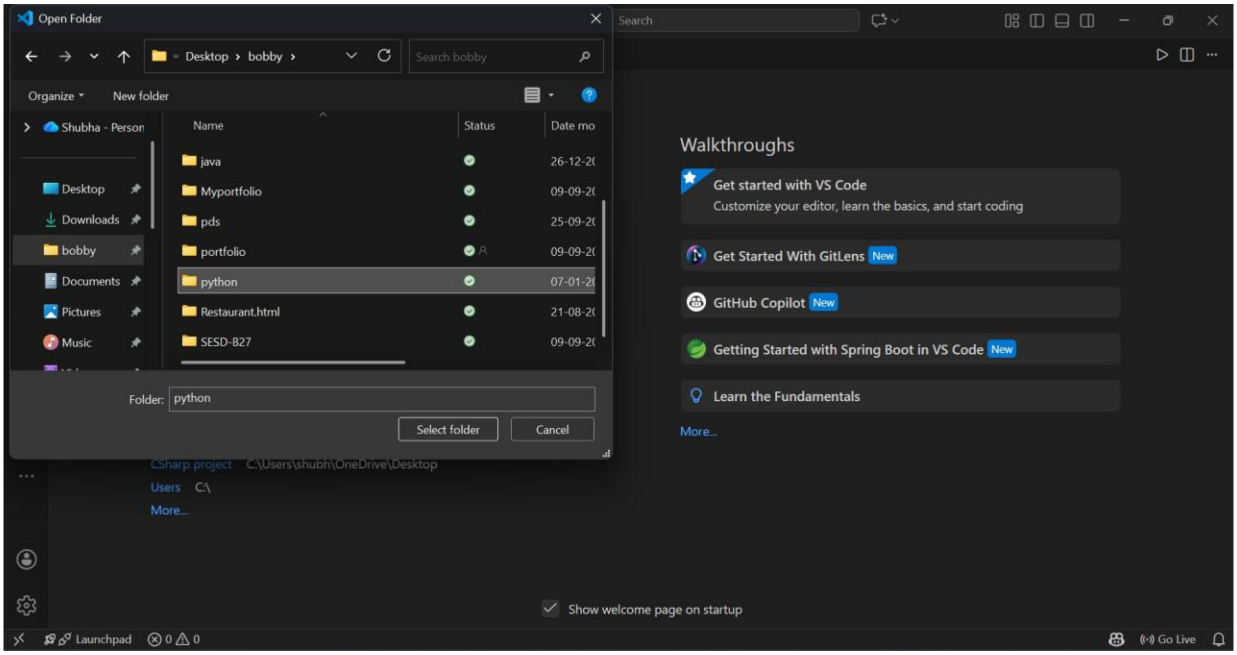
1) Script (reads two numbers from the user) 🔧

```
# sum_script.py
a = float(input("Enter first numl
b = float(input("Enter second nur
print("Sum:", a + b)
```

2) Function (reusable) 💡

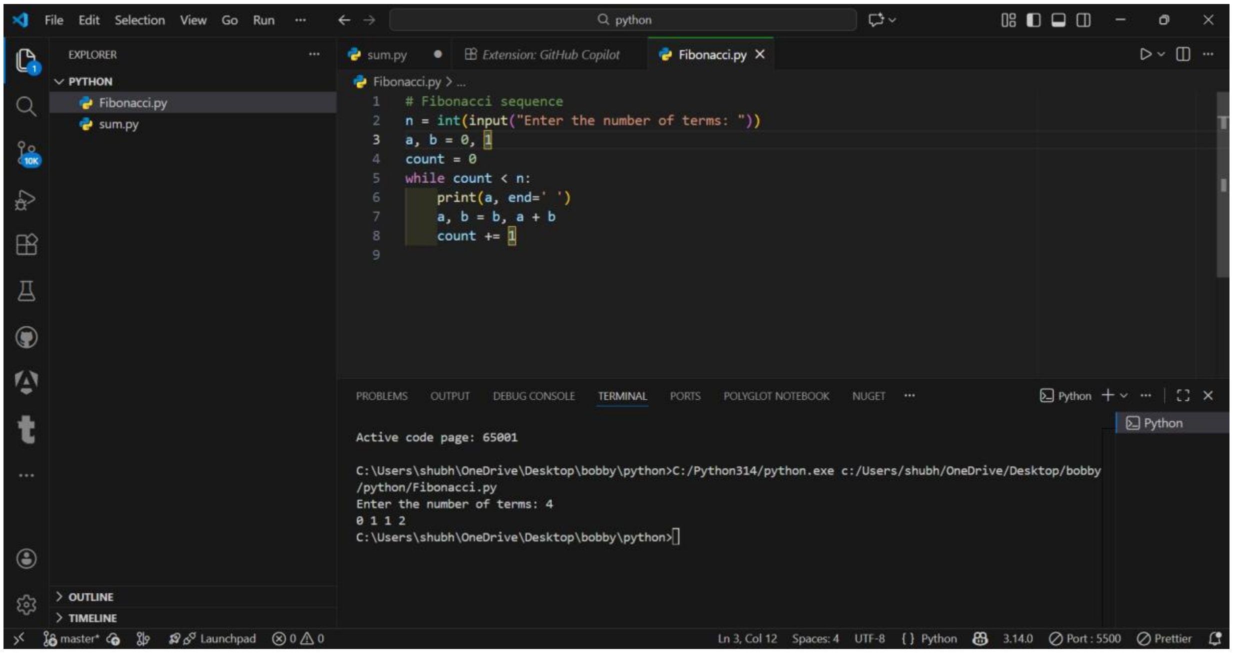
```
def add(a, b):
    return a + b

# usage
print(add(3, 4))  # prints 7
```

Note: use `int()` instead of `float()` if you only want integers.



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

```
# Fibonacci sequence
```


File Edit Selection View Go Run
python
EXPLORER
PYTHON
Fibonacci.py
sum.py
sum.py
Extension: GitHub Copilot
Fibonacci.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK NUGET
Python
master*
Launchpad
Ln 3, Col 12 Spaces: 4 UTF-8 { } Python 3.14.0 Port : 5500 Prettier

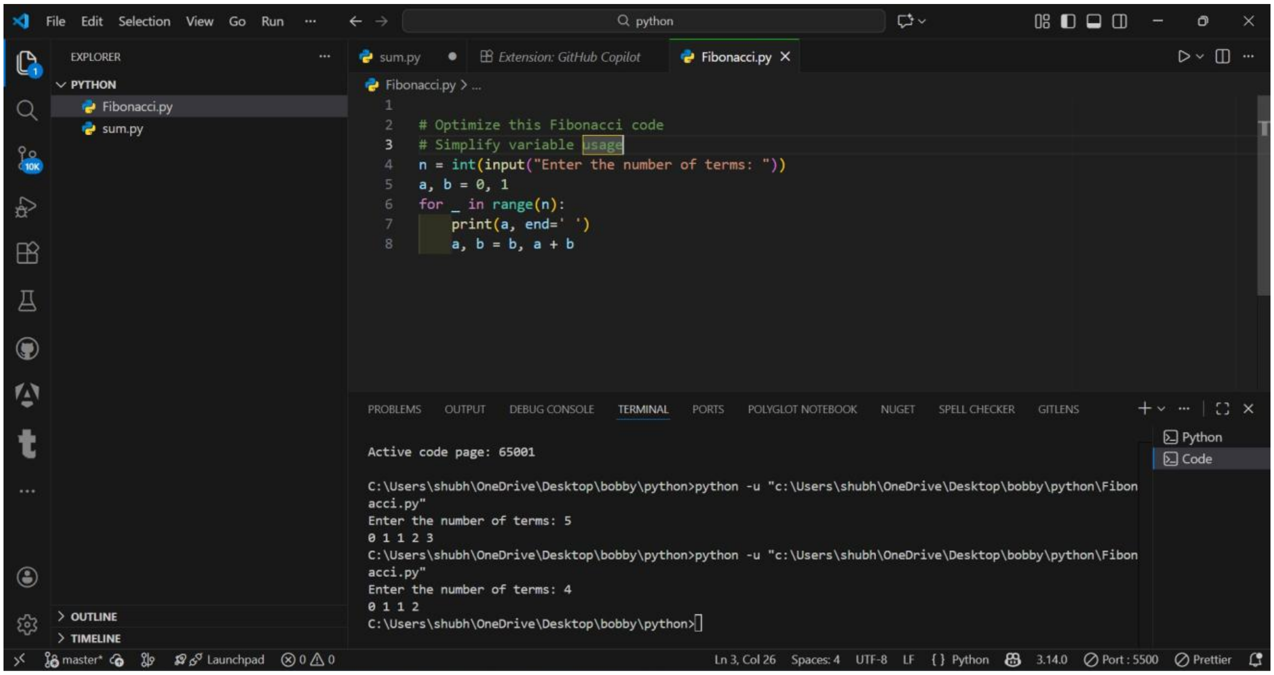
```
# Fibonacci sequence
n = int(input("Enter the number of terms: "))
a, b = 0, 1
count = 0
while count < n:
    print(a, end=' ')
    a, b = b, a + b
    count += 1
```

```
Active code page: 65001

C:\Users\shubh\OneDrive\Desktop\bobby\python>C:/Python314/python.exe c:/Users/shubh/OneDrive/Desktop/bobby
/python/Fibonacci.py
Enter the number of terms: 4
0 1 1 2
C:\Users\shubh\OneDrive\Desktop\bobby\python>
```

Task 2: AI Code Optimization & Cleanup :

Optimize this Fibonacci code

Simplify variable usage


File Edit Selection View Go Run ···
python
EXPLORER
PYTHON
Fibonacci.py
sum.py
sum.py
Extension: GitHub Copilot
Fibonacci.py
Fibonacci.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK NUGET SPELL CHECKER GITLENS
Python
Code
OUTLINE
TIMELINE
master* Launchpad ⊗ 0 ⚠ 0
Ln 3, Col 26 Spaces: 4 UTF-8 LF { } Python 3.14.0 Port : 5500 Prettier

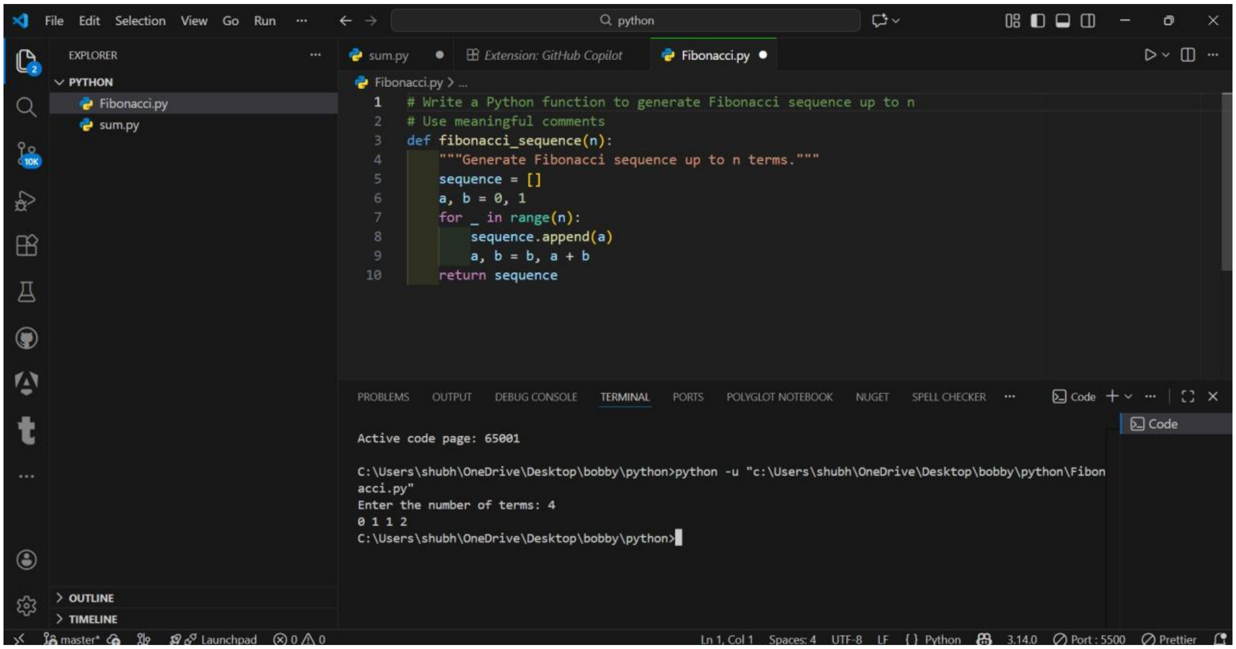
```python

# Optimize this Fibonacci code
# Simplify variable usage
n = int(input("Enter the number of terms: "))
a, b = 0, 1
for _ in range(n):
    print(a, end=' ')
    a, b = b, a + b
```

```
Active code page: 65001

C:\Users\shubh\OneDrive\Desktop\bobby\python>python -u "c:\Users\shubh\OneDrive\Desktop\bobby\python\Fibon
acci.py"
Enter the number of terms: 5
0 1 1 2 3
C:\Users\shubh\OneDrive\Desktop\bobby\python>python -u "c:\Users\shubh\OneDrive\Desktop\bobby\python\Fibon
acci.py"
Enter the number of terms: 4
0 1 1 2
C:\Users\shubh\OneDrive\Desktop\bobby\python>
```

Task 3: Modular Design Using AI Assistance (Function-Based Fibonacci) :

Write a Python function to generate Fibonacci sequence up to n

Use meaningful comments


File Edit Selection View Go Run ···
python
EXPLORER
PYTHON
Fibonacci.py
sum.py
sum.py
Extension: GitHub Copilot
Fibonacci.py
Fibonacci.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK NUGET SPELL CHECKER ···
Code
Code
OUTLINE
TIMELINE
master* Launchpad ⊗ 0 ⚠ 0
Ln 1, Col 1 Spaces: 4 UTF-8 LF { } Python 3.14.0 Port : 5500 Prettier

```python
# Write a Python function to generate Fibonacci sequence up to n
# Use meaningful comments
def fibonacci_sequence(n):
    """Generate Fibonacci sequence up to n terms."""
    sequence = []
    a, b = 0, 1
    for _ in range(n):
        sequence.append(a)
        a, b = b, a + b
    return sequence
```

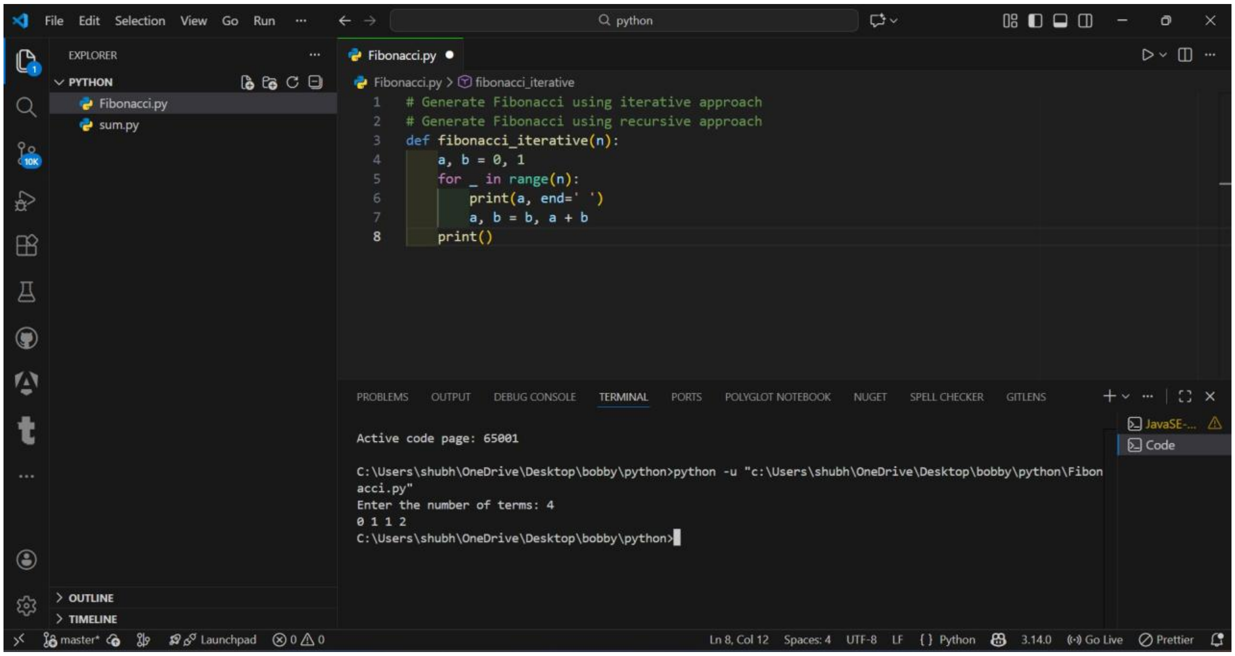
```
Active code page: 65001

C:\Users\shubh\OneDrive\Desktop\bobby\python>python -u "c:\Users\shubh\OneDrive\Desktop\bobby\python\Fibon
acci.py"
Enter the number of terms: 4
0 1 1 2
C:\Users\shubh\OneDrive\Desktop\bobby\python>
```

Task 4: Comparative Analysis – Procedural vs Modular Code

Criteria	Without Functions	With Functions
Code Clarity	Lower	Higher
Reusability	No	Yes
Debugging	Harder	Easier
Scalability	Poor	Excellent
Suitable for Large Systems	No	Yes

Task 5: Iterative vs Recursive Fibonacci (AI-Generated):

Generate Fibonacci using iterative approach

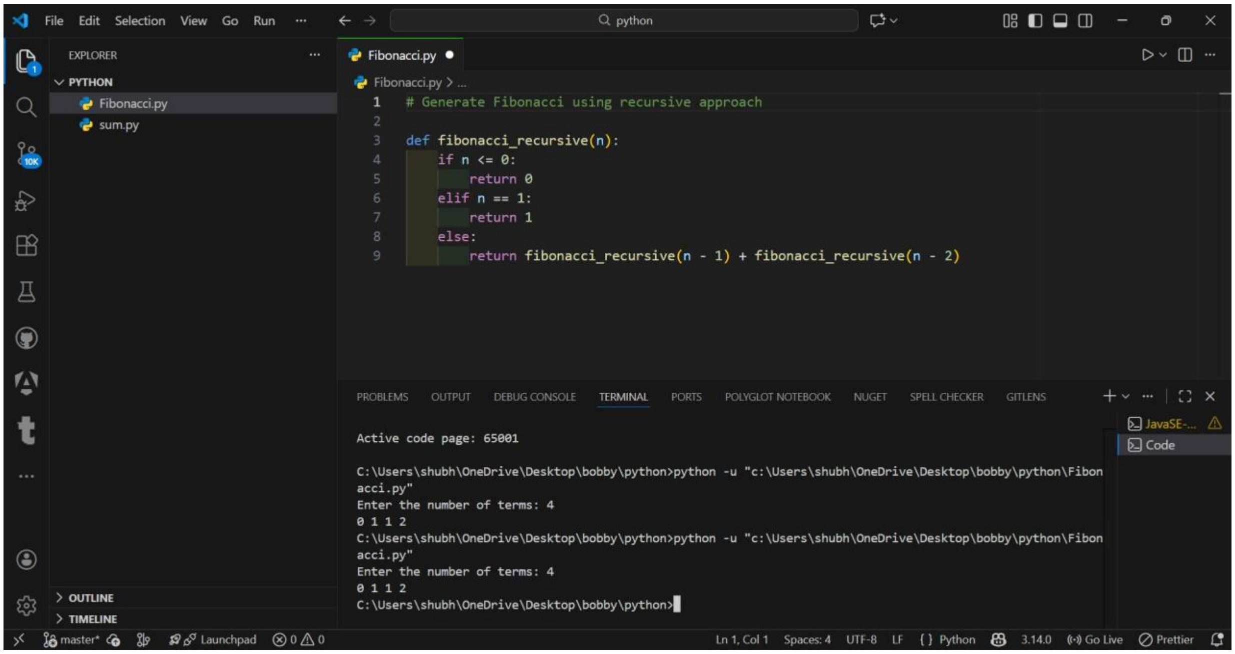


File Edit Selection View Go Run ···
python
EXPLORER
PYTHON
Fibonacci.py
sum.py
Fibonacci.py
Fibonacci.py > fibonacci_iterative
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POLYGLOT NOTEBOOK NUGET SPELL CHECKER GITLENS
JavaSE-...
Code
OUTLINE
TIMELINE
master* Launchpad ⊗ 0 ⚠ 0
Ln 8, Col 12 Spaces: 4 UTF-8 LF { } Python 3.14.0 Go Live Prettier

```python
# Generate Fibonacci using iterative approach
# Generate Fibonacci using recursive approach
def fibonacci_iterative(n):
    a, b = 0, 1
    for _ in range(n):
        print(a, end=' ')
        a, b = b, a + b
    print()
```

```
Active code page: 65001

C:\Users\shubh\OneDrive\Desktop\bobby\python>python -u "c:\Users\shubh\OneDrive\Desktop\bobby\python\Fibon
acci.py"
Enter the number of terms: 4
0 1 1 2
C:\Users\shubh\OneDrive\Desktop\bobby\python>
```

Generate Fibonacci using recursive approach



```python
# Generate Fibonacci using recursive approach

def fibonacci_recursive(n):
    if n <= 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci_recursive(n - 1) + fibonacci_recursive(n - 2)
```

```
Active code page: 65001

C:\Users\shubh\OneDrive\Desktop\bobby\python>python -u "c:\Users\shubh\OneDrive\Desktop\bobby\python\Fibon
acci.py"
Enter the number of terms: 4
0 1 1 2
C:\Users\shubh\OneDrive\Desktop\bobby\python>python -u "c:\Users\shubh\OneDrive\Desktop\bobby\python\Fibon
acci.py"
Enter the number of terms: 4
0 1 1 2
C:\Users\shubh\OneDrive\Desktop\bobby\python>
```