Course Title: AI Assisted Coding

Course Code: 23CS002PC304

Faculty Name: Dr. R. Prashant Kumar

Name: B. Harshini

HT no: 2303A52242- Batch(36)

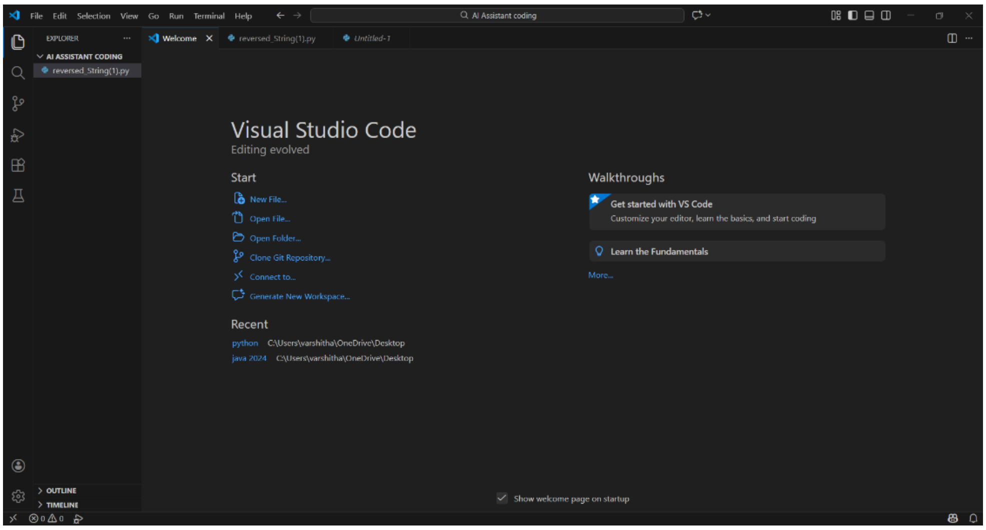
Question:

Lab 1: Environment Setup – GitHub Copilot and VS Code Integration + Understanding AI-assisted Coding Workflow

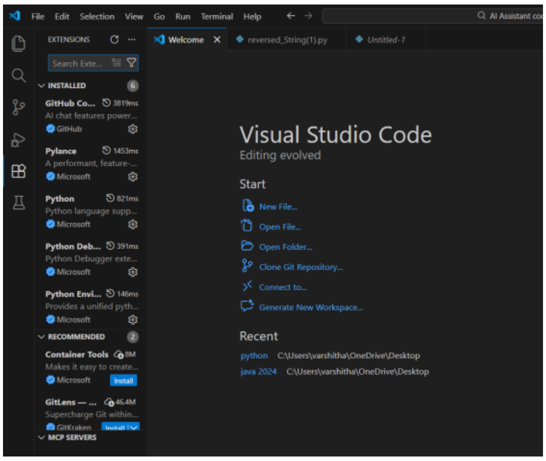
Task 0:

- Install and configure GitHub Copilot in VS Code. Take screenshots of each step.

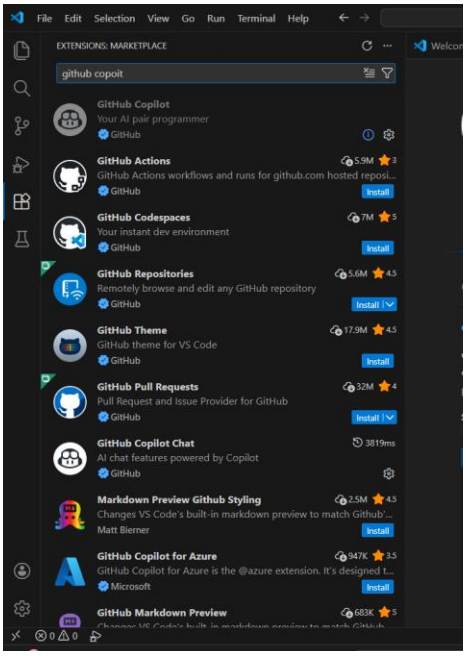
Step 1: Open Visual Studio Code



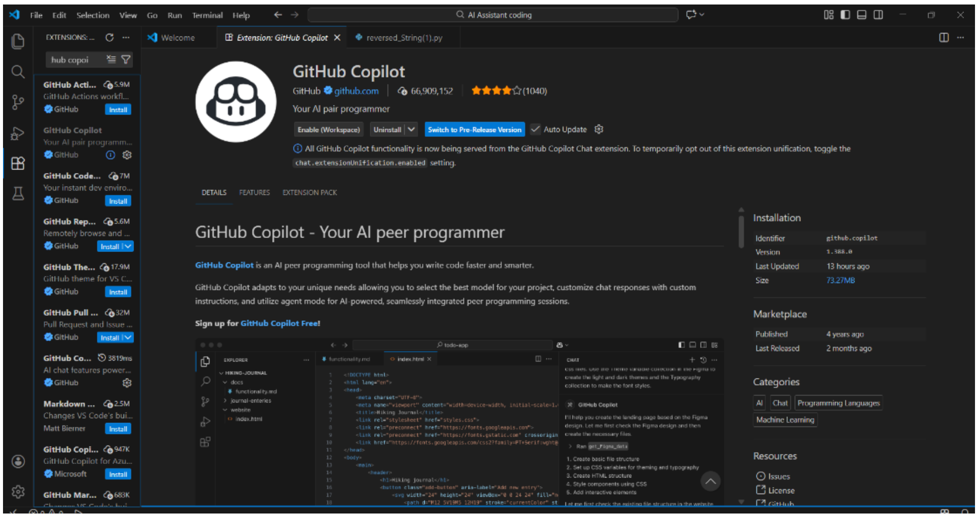
Step 2: Open Extensions Panel



Step 3: Search for GitHub Copilot


File Edit Selection View Go Run Terminal Help
EXTENSIONS: MARKETPLACE
github copoit
GitHub Copilot
Your AI pair programmer
GitHub
GitHub Actions
GitHub Actions workflows and runs for github.com hosted reposi...
GitHub Codespaces
Your instant dev environment
GitHub Repositories
Remotely browse and edit any GitHub repository
GitHub Theme
GitHub theme for VS Code
GitHub Pull Requests
Pull Request and Issue Provider for GitHub
GitHub Copilot Chat
AI chat features powered by Copilot
Markdown Preview Github Styling
Changes VS Code's built-in markdown preview to match Github'...
Matt Bierner
GitHub Copilot for Azure
GitHub Copilot for Azure is the @azure extension. It's designed t...
Microsoft
GitHub Markdown Preview
Install

Step 4: Install GitHub Copilot



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!

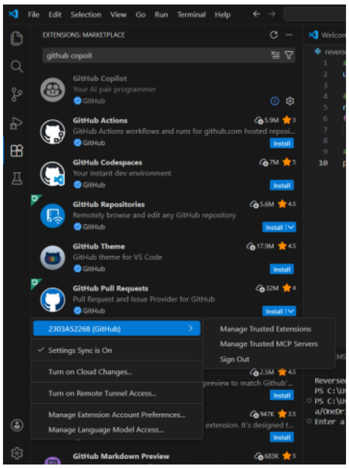
Installation

Identifier	github.copilot
Version	1.388.0
Last Updated	13 hours ago
Size	73.27MB

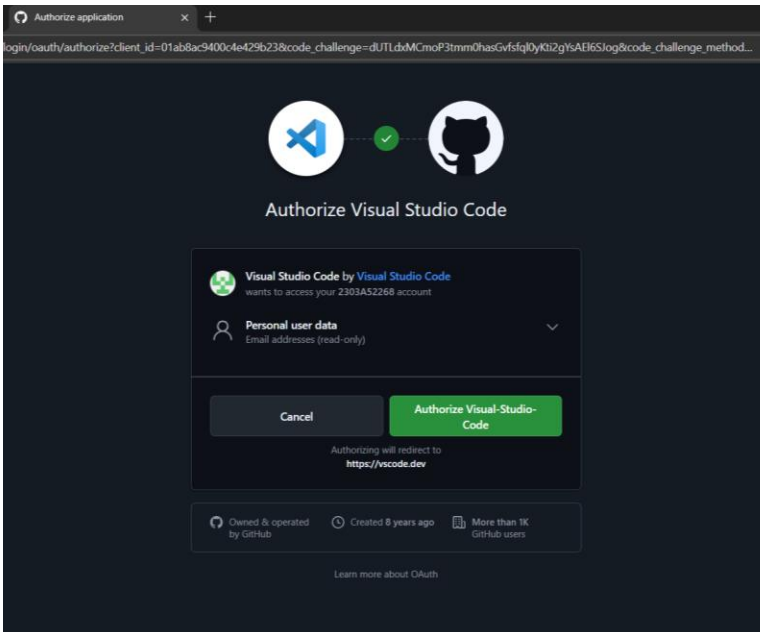
Marketplace

Published	4 years ago
Last Released	2 months ago

Step 5: Sign in to GitHub Account



Step 6: Authorize GitHub Copilot


Authorize application
login/oauth/authorize?client_id=01ab8ac9400c4e429b23&code_challenge=dUTLdxMCmoP3tmm0hasGvfsfql0yKti2gYsAEl6SJog&code_challenge_method...

Authorize Visual Studio Code

Visual Studio Code by Visual Studio Code
wants to access your 2303A52268 account

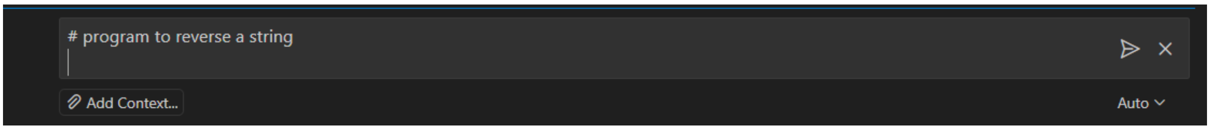
Personal user data
Email addresses (read-only)

Cancel
Authorize Visual-Studio-Code

Authorizing will redirect to
https://vscode.dev

Owned & operated by GitHub
Created 8 years ago
More than 1K GitHub users
Learn more about OAuth

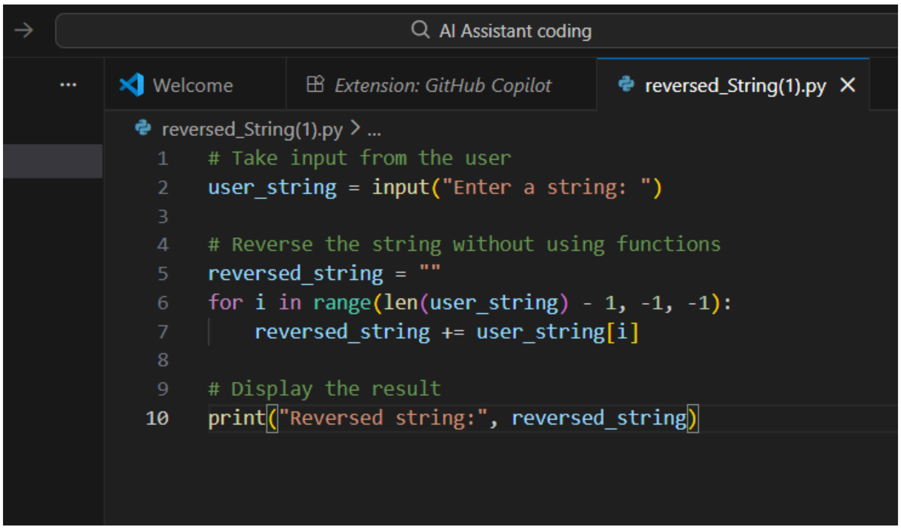
Step 7: Verify Copilot is Enabled



```
# program to reverse a string
```

Task 1: AI-Generated Logic Without Modularization (String Reversal Without Functions)

program to reverse a string

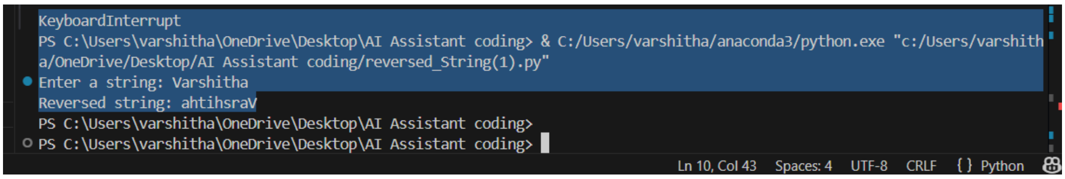


```python
# Take input from the user
user_string = input("Enter a string: ")

# Reverse the string without using functions
reversed_string = ""
for i in range(len(user_string) - 1, -1, -1):
    reversed_string += user_string[i]

# Display the result
print("Reversed string:", reversed_string)
```

Output:



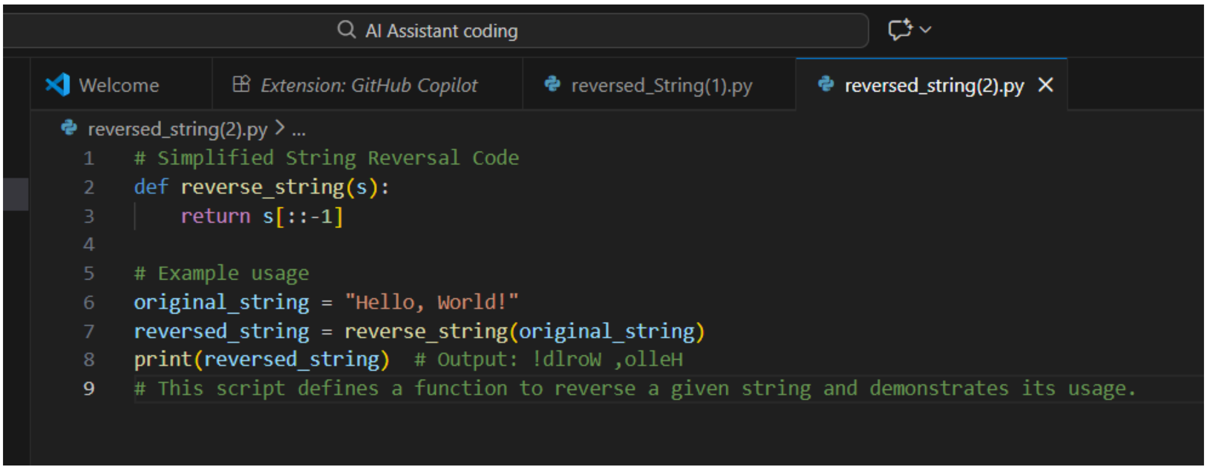
```
KeyboardInterrupt
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> & C:/Users/varshitha/anaconda3/python.exe "c:/Users/varshith
a/OneDrive/Desktop/AI Assistant coding/reversed_String(1).py"
Enter a string: Varshitha
Reversed string: ahtihsraV
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding>
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding>
```

Explanation

- The input() function takes a string from the user.
- An empty string rev is created to store the reversed result.
- The for loop iterates through the string from the last character to the first.
- Each character is appended to rev.
- The final reversed string is printed.
- The logic is written directly in the main code without using functions

Task 2: Efficiency & Logic Optimization (Readability Improvement)

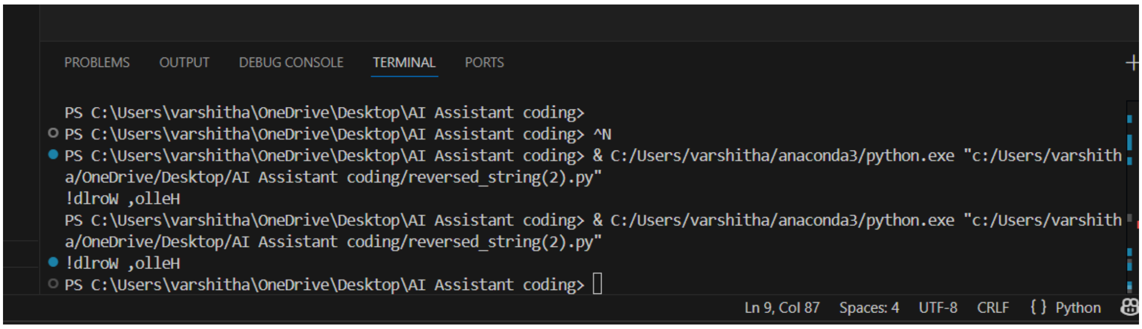
Simplified String Reversal Code



```python
# Simplified String Reversal Code
def reverse_string(s):
    return s[::-1]

# Example usage
original_string = "Hello, World!"
reversed_string = reverse_string(original_string)
print(reversed_string)  # Output: !dlroW ,olleH
# This script defines a function to reverse a given string and demonstrates its usage.
```

Output:



```
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding>
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> ^N
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> & C:/Users/varshitha/anaconda3/python.exe "c:/Users/varshith
a/OneDrive/Desktop/AI Assistant coding/reversed_string(2).py"
!dlroW ,olleH
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> & C:/Users/varshitha/anaconda3/python.exe "c:/Users/varshith
a/OneDrive/Desktop/AI Assistant coding/reversed_string(2).py"
!dlroW ,olleH
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> 
```

Explanation of Optimization

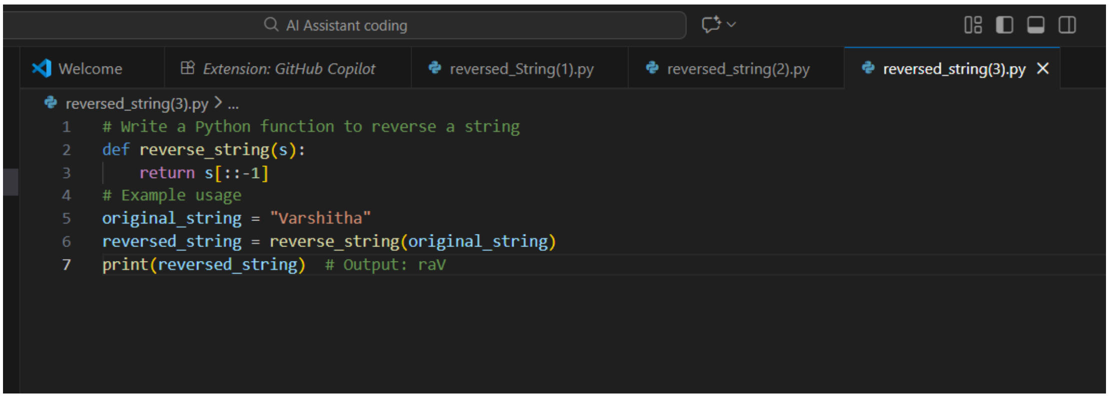
- The loop and extra variable were removed
- Python slicing reverses the string in a single step
- Code is shorter, cleaner, and easier to understand

Time Complexity Explanation

- Original code: **O(n)** (manual loop)
- Optimized code: **O(n)** (built-in slicing)
- Although complexity remains the same, slicing is **faster in practice** due to internal optimization

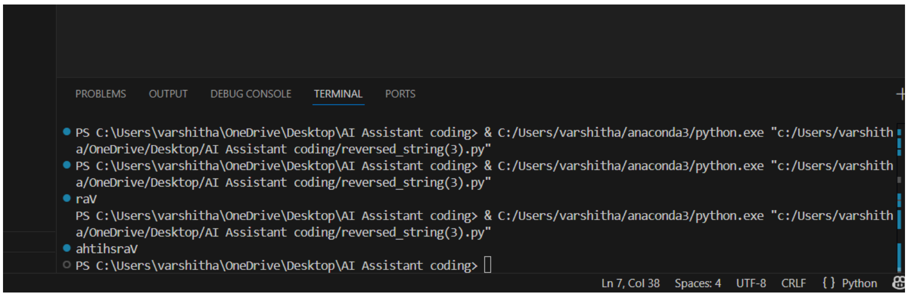
Task 3: Modular Design Using AI Assistance (String Reversal Using Functions)

Write a Python function to reverse a string



```python
# Write a Python function to reverse a string
def reverse_string(s):
    return s[::-1]
# Example usage
original_string = "Varshitha"
reversed_string = reverse_string(original_string)
print(reversed_string)  # Output: raV
```

Output:



```
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> & C:/Users/varshitha/anaconda3/python.exe "c:/Users/varshith
a/OneDrive/Desktop/AI Assistant coding/reversed_string(3).py"
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> & C:/Users/varshitha/anaconda3/python.exe "c:/Users/varshith
a/OneDrive/Desktop/AI Assistant coding/reversed_string(3).py"
raV
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> & C:/Users/varshitha/anaconda3/python.exe "c:/Users/varshith
a/OneDrive/Desktop/AI Assistant coding/reversed_string(3).py"
ahtihsraV
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> 
```

Explanation

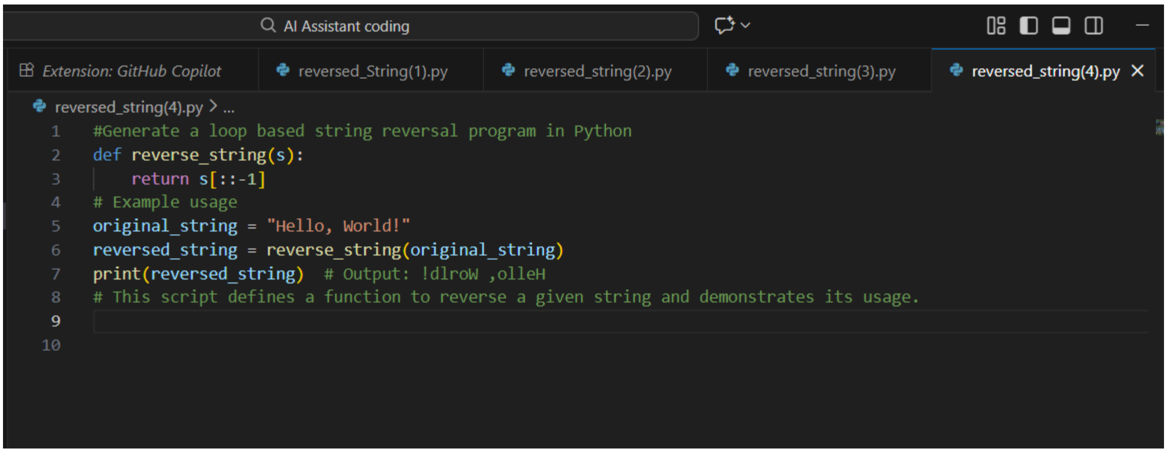
- A function reverse_string() is defined to reverse a string.
- The function takes one parameter text.
- The slicing method [::-1] is used to reverse the string.
- The reversed string is returned to the caller.
- User input is passed to the function.
- The result is printed.
- This modular approach improves reusability and readability.

Task 4: Comparative Analysis – Procedural vs Modular Approach (With vs Without Functions)

Aspect	Without Function (Procedural)	With Function (Modular)
Code Clarity	Moderate	High
Reusability	Not reusable	Highly reusable
Debugging	Difficult	Easier
Maintainability	Low	High
Large-scale Suitability	Poor	Good

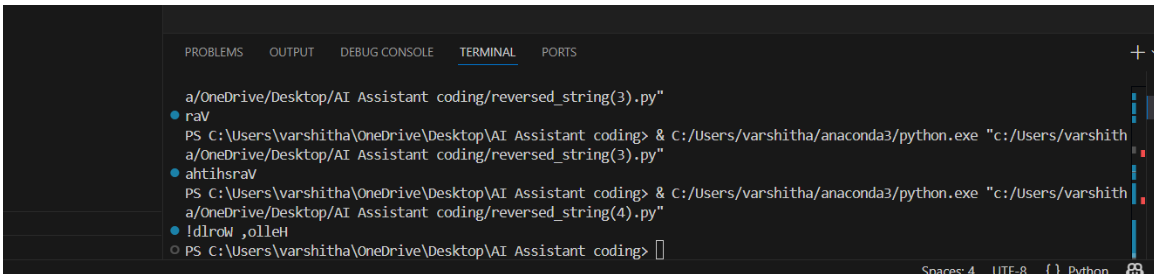
Task 5: AI-Generated Iterative vs Recursive Fibonacci Approaches (Different Algorithmic Approaches to String Reversal)

#Generate a loop based string reversal program in Python



```python
#Generate a loop based string reversal program in Python
def reverse_string(s):
    return s[::-1]
# Example usage
original_string = "Hello, World!"
reversed_string = reverse_string(original_string)
print(reversed_string)  # Output: !dlroW ,olleH
# This script defines a function to reverse a given string and demonstrates its usage.
```

Output:



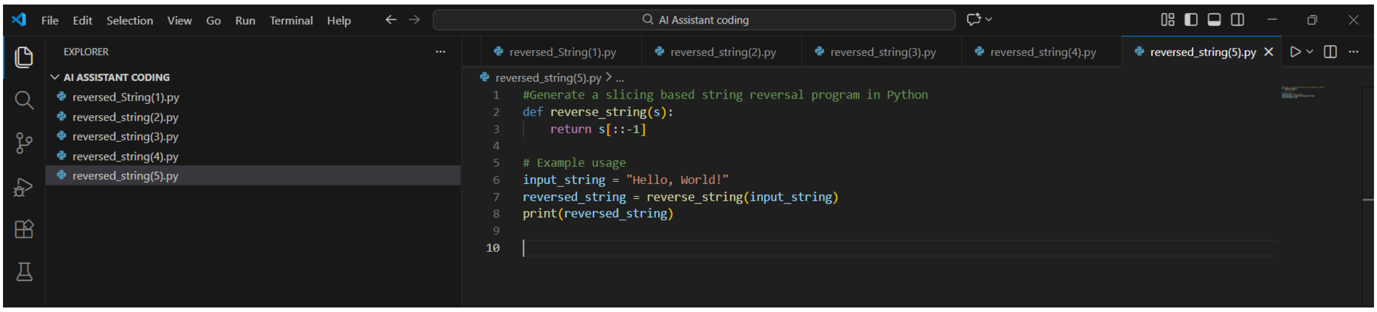
```
a/OneDrive/Desktop/AI Assistant coding/reversed_string(3).py"
raV
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> & C:/Users/varshitha/anaconda3/python.exe "c:/Users/varshith
a/OneDrive/Desktop/AI Assistant coding/reversed_string(3).py"
ahtihsraV
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> & C:/Users/varshitha/anaconda3/python.exe "c:/Users/varshith
a/OneDrive/Desktop/AI Assistant coding/reversed_string(4).py"
!dlroW ,olleH
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> 
```

Explanation

- The user inputs a string.
- An empty string rev is created.
- The loop reads each character from left to right.
- Each character is added at the beginning of rev, reversing the order.

- The reversed string is printed.
- This method helps understand string manipulation logic.

#Generate a slicing based string reversal program in Python

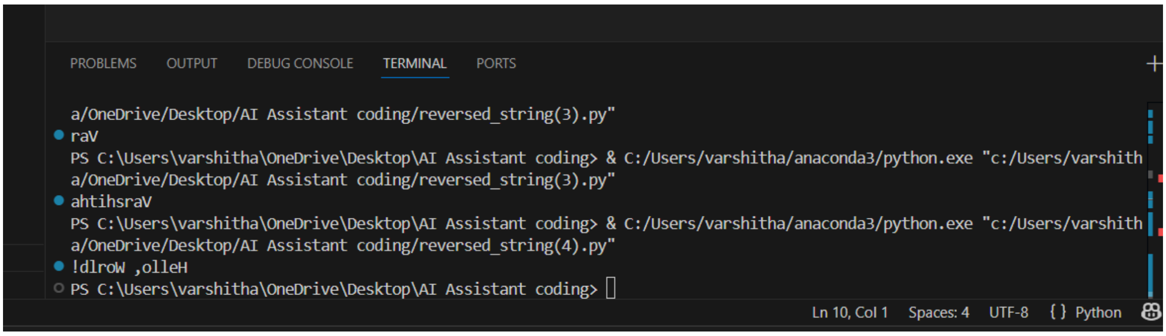


```python
#Generate a slicing based string reversal program in Python
def reverse_string(s):
    return s[::-1]

# Example usage
input_string = "Hello, World!"
reversed_string = reverse_string(input_string)
print(reversed_string)

```

Output:


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ln 10, Col 1 Spaces: 4 UTF-8 { } Python

```
a/OneDrive/Desktop/AI Assistant coding/reversed_string(3).py"
raV
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> & C:/Users/varshitha/anaconda3/python.exe "c:/Users/varshith
a/OneDrive/Desktop/AI Assistant coding/reversed_string(3).py"
ahtihsraV
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> & C:/Users/varshitha/anaconda3/python.exe "c:/Users/varshith
a/OneDrive/Desktop/AI Assistant coding/reversed_string(4).py"
!dlroW ,olleH
PS C:\Users\varshitha\OneDrive\Desktop\AI Assistant coding> 
```

Explanation

- The string is taken from the user.
- Python slicing reverses the string efficiently.
- The reversed string is printed directly.
- This approach is best for large inputs and real-world applications.

Comparison of Approaches

Aspect	**Loop-Based**	**Slicing-Based**
Execution Flow	Step-by-step reversal	Single operation
Time Complexity	O(n)	O(n)
Performance for Large Inputs	Slower	Faster
Readability	Moderate	Very High
Best Usage	Learning logic	Production code