

Course Title: AI Assisted Coding

Course Code: 23CS002PC304

Faculty Name: Dr. R. Prashant Kumar

Name: Vyshnavi K

HT no: 2303A52273- Batch(36)

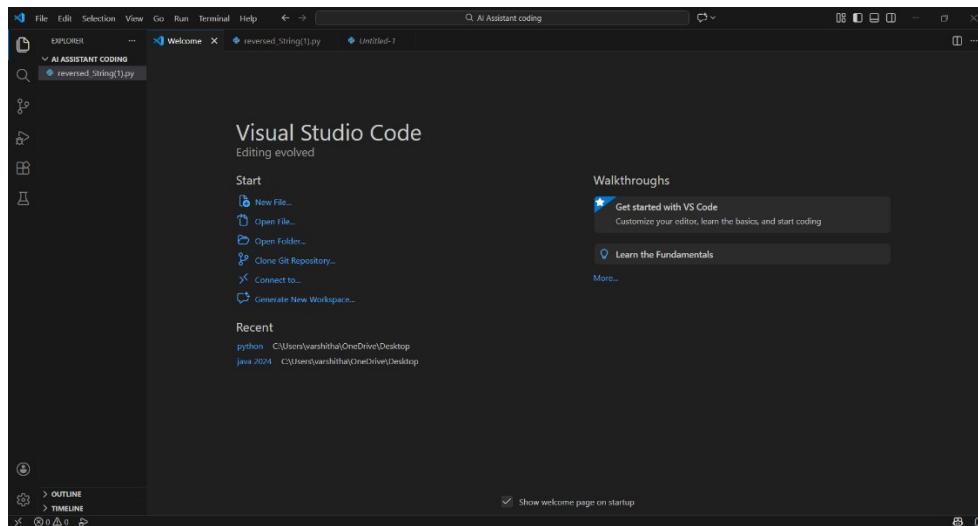
Question:

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

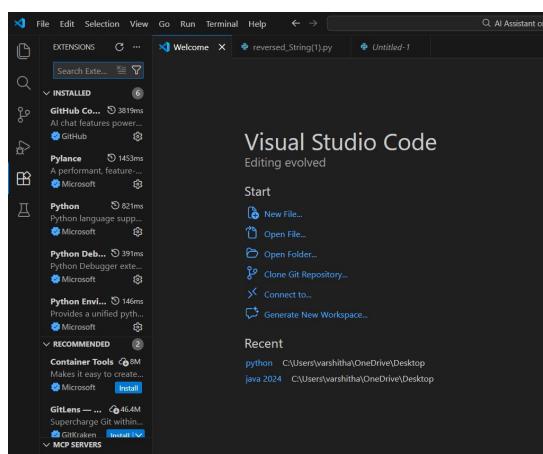
Task 0:

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

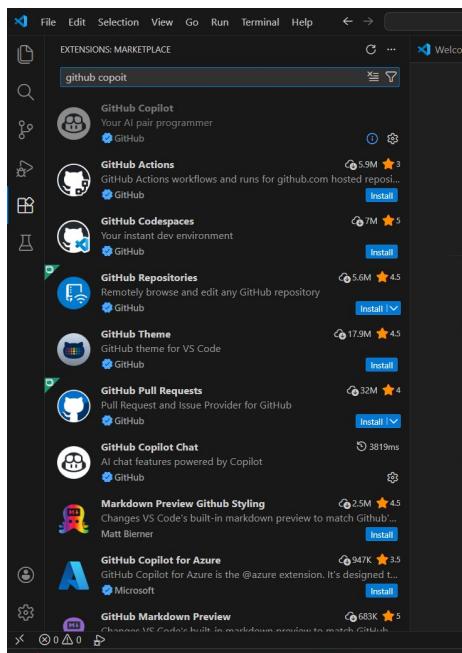
Step 1: Open Visual Studio Code



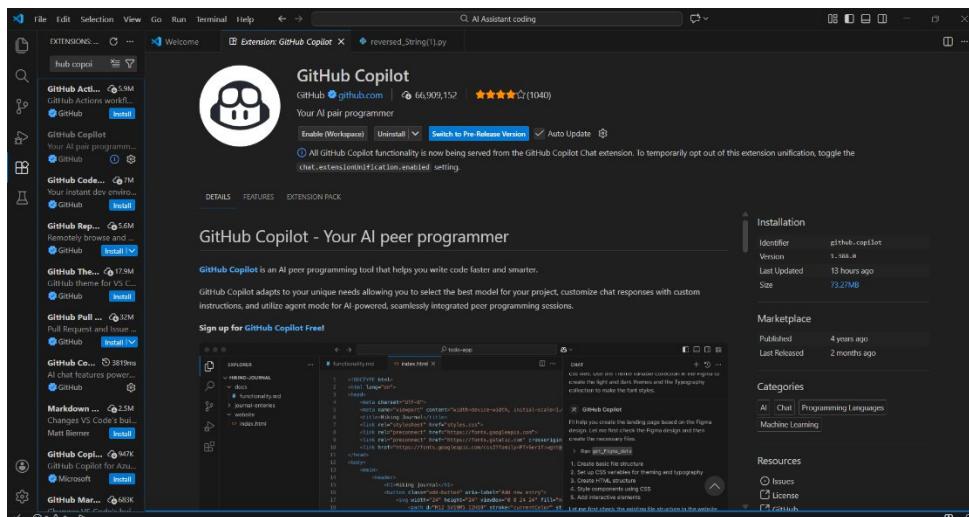
Step 2: Open Extensions Panel



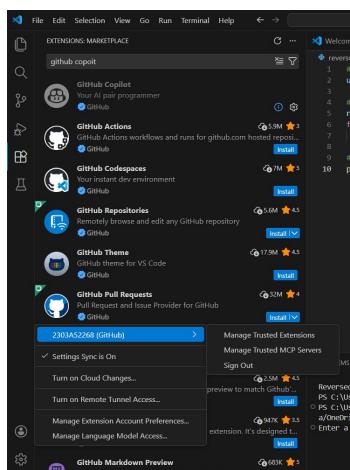
Step 3: Search for GitHub Copilot



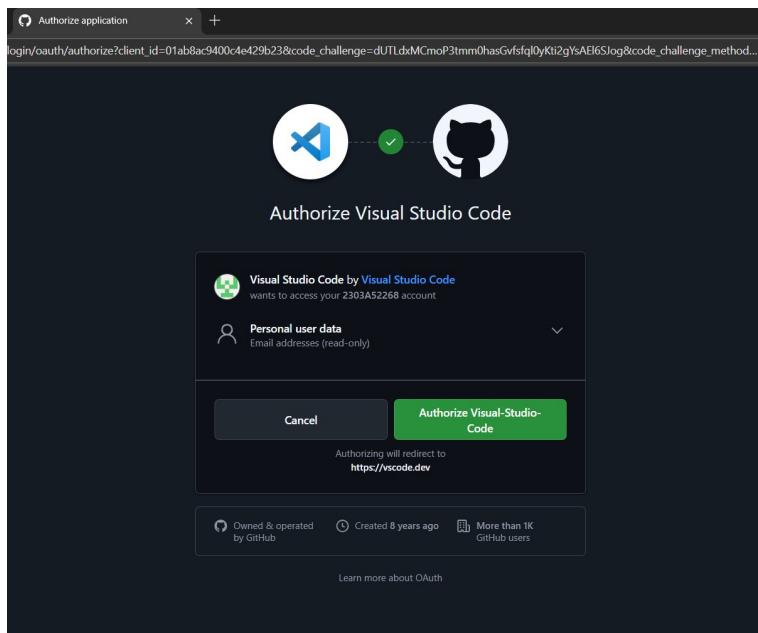
Step 4: Install GitHub Copilot



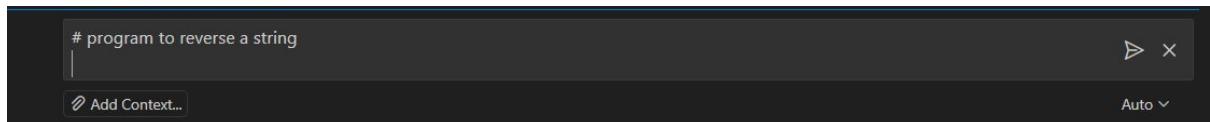
Step 5: Sign in to GitHub Account



Step 6: Authorize GitHub Copilot

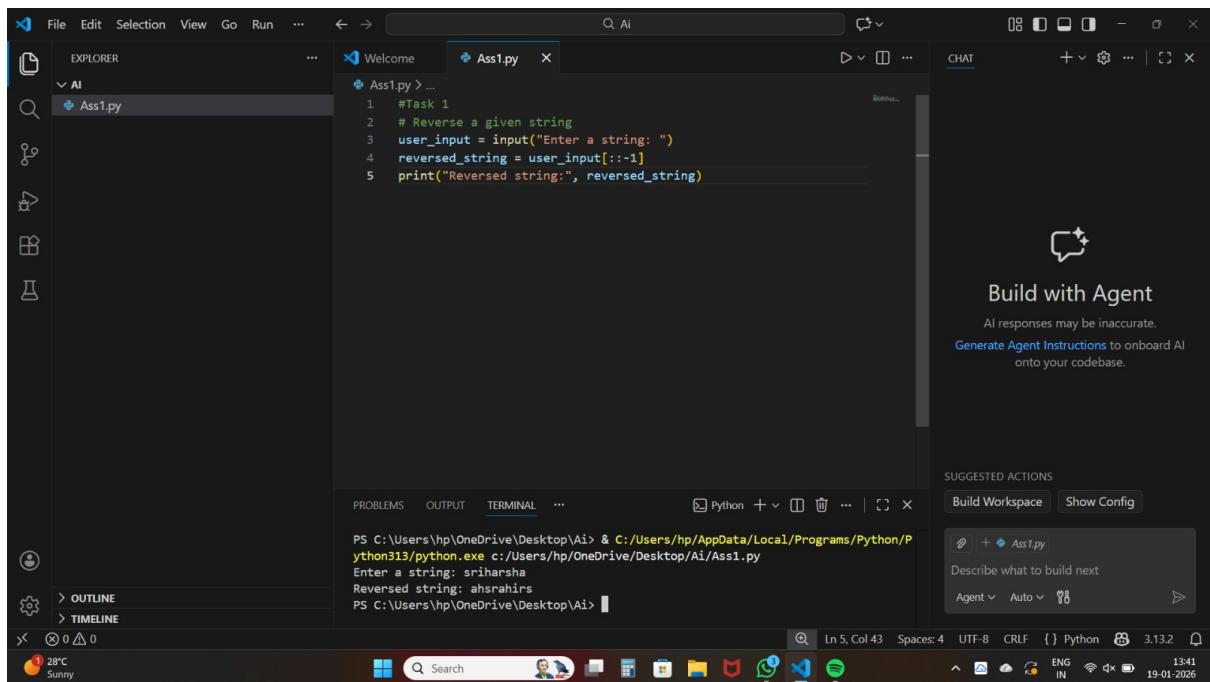


Step 7: Verify Copilot is Enabled



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

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

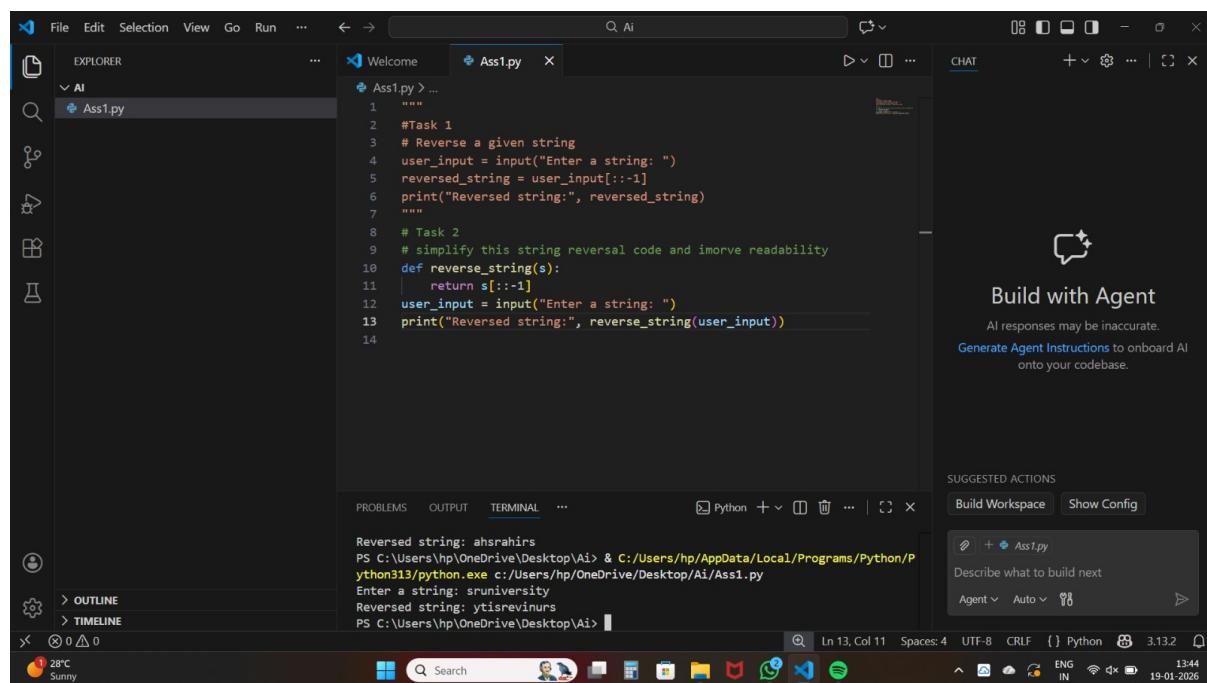


Explanation

- 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



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

- Explorer View:** Shows a file named `Ass1.py` in the AI folder.
- Code Editor:** Displays the following Python code:

```

1 """
2 #Task 1
3 # Reverse a given string
4 user_input = input("Enter a string: ")
5 reversed_string = user_input[::-1]
6 print("Reversed string:", reversed_string)
7 """
8 # Task 2
9 # simplify this string reversal code and improve readability
10 def reverse_string(s):
11     return s[::-1]
12 user_input = input("Enter a string: ")
13 print("Reversed string:", reverse_string(user_input))
14

```
- Terminal:** Shows the output of running the code:

```

Reversed string: ahshahris
PS C:\Users\hp\OneDrive\Desktop\Ai> & C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe c:/Users/hp/OneDrive/Desktop/Ai/Ass1.py
Enter a string: suniversity
Reversed string: ytisrevinurs
PS C:\Users\hp\OneDrive\Desktop\Ai>

```
- Suggested Actions:** A sidebar on the right suggests "Build with Agent" and "Build Workspace".
- Bottom Status Bar:** Shows the current weather (28°C, Sunny), search bar, and system status.

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

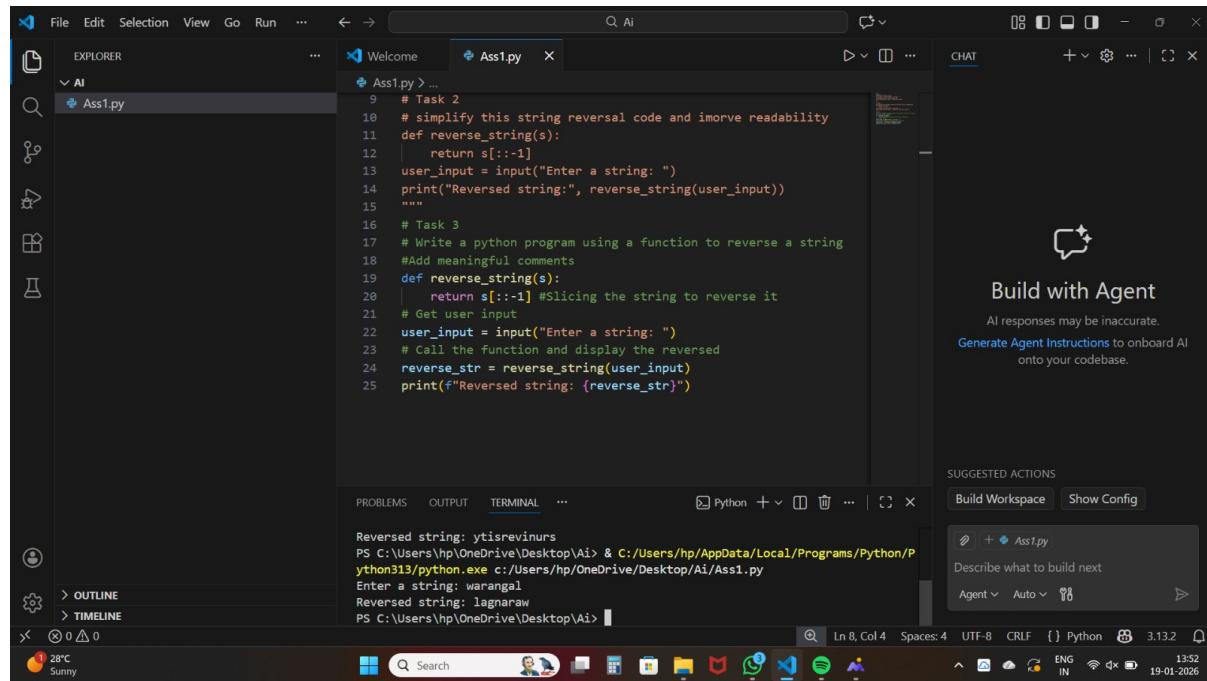
Time Complexity Explanation

- 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



The screenshot shows a dark-themed instance of Visual Studio Code. In the center, the code editor displays a file named 'Ass1.py' containing the following Python code:

```

# Task 2
# simplify this string reversal code and improve readability
def reverse_string(s):
    return s[::-1]
user_input = input("Enter a string: ")
print("Reversed string:", reverse_string(user_input))

# Task 3
# Write a python program using a function to reverse a string
# Add meaningful comments
def reverse_string(s):
    return s[::-1] #Slicing the string to reverse it
# Get user input
user_input = input("Enter a string: ")
# Call the function and display the reversed
reverse_str = reverse_string(user_input)
print(f"Reversed string: {reverse_str}")

```

To the right of the code editor, there's an 'AI' sidebar with a 'Build with Agent' section. It includes a button to 'Generate Agent Instructions' and a note that 'AI responses may be inaccurate'. Below the code editor, the terminal window shows the output of running the script:

```

Reversed string: ytisrevinurus
PS C:\Users\hp\OneDrive\Desktop\Ai & C:/Users/hp/AppData/Local/Programs/Python/Python311/python.exe c:/Users/hp/OneDrive/Desktop/Ai/Ass1.py
Enter a string: warangal
Reversed string: lagnaraw
PS C:\Users\hp\OneDrive\Desktop\Ai>

```

The status bar at the bottom indicates the file is saved, the line number is 8, column is 4, spaces are 4, encoding is UTF-8, and the date is 19-01-2026.

Explanation

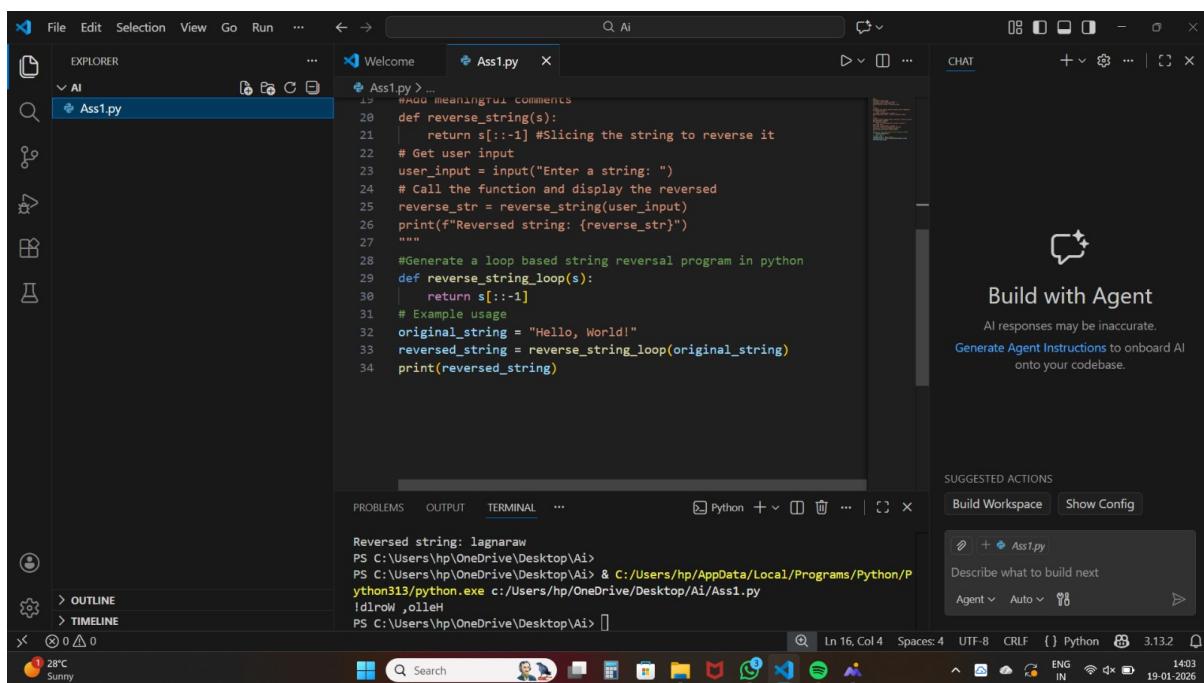
- A function `reverse_string()` is defined to reverse a string.
- The function takes one parameter `text`.
- The slicing method `[::-1]` is used to reverse the string.
- The reversed string is returned to the caller.
- User input is passed to the function.
- The result is printed.
- This modular approach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



The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows a file named "Ass1.py" selected.
- Code Editor:** Displays the following Python code:

```

1 #Generate a loop based string reversal program in python
2 def reverse_string(s):
3     return s[::-1] #Slicing the string to reverse it
4 # Get user input
5 user_input = input("Enter a string: ")
6 # Call the function and display the reversed
7 reverse_str = reverse_string(user_input)
8 print(f"Reversed string: {reverse_str}")
9 """
10 #Generate a loop based string reversal program in python
11 def reverse_string_loop(s):
12     return s[::-1]
13 # Example usage
14 original_string = "Hello, World!"
15 reversed_string = reverse_string_loop(original_string)
16 print(reversed_string)
17

```
- Terminal:** Shows the command `python Ass1.py` being run, with the output "Reversed string: !dlroW ,olleH".
- AI Panel:** On the right, there is an "AI" tab with a "Build with Agent" section. It says "AI responses may be inaccurate." and "Generate Agent Instructions to onboard AI onto your codebase."
- Suggested Actions:** A panel at the bottom right suggests "Build Workspace" and "Show Config".

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
```

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar has sections for EXPLORER, AI, and OUTLINE. The AI section is expanded, showing a file named 'Ass1.py'. The main editor area displays the following Python code:

```
1 #Add meaningful comments
2 def reverse_string(s):
3     return s[::-1] #Slicing the string to reverse it
4 # Get user input
5 user_input = input("Enter a string: ")
6 # Call the function and display the reversed
7 reverse_str = reverse_string(user_input)
8 print(f"Reversed string: {reverse_str}")
9 """
10 #Generate a loop based string reversal program in python
11 def reverse_string_loop(s):
12     return s[::-1]
13 # Example usage
14 original_string = "Hello, World!"
15 reversed_string = reverse_string_loop(original_string)
16 print(reversed_string)
```

The bottom terminal pane shows the output of running the script:

```
Reversed string: lagnaraw
PS C:\Users\hp\OneDrive\Desktop\AI>
PS C:\Users\hp\OneDrive\Desktop\AI> & C:/Users/hp/AppData/Local/Programs/Python/Python313/python.exe c:/Users/hp/OneDrive/Desktop/AI/Ass1.py
!dir oW ,olleH
PS C:\Users\hp\OneDrive\Desktop\AI>
```

The status bar at the bottom right indicates the file has 16 lines, 4 spaces, and is in UTF-8 format, with a timestamp of 14:03 on 19-01-2026.

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