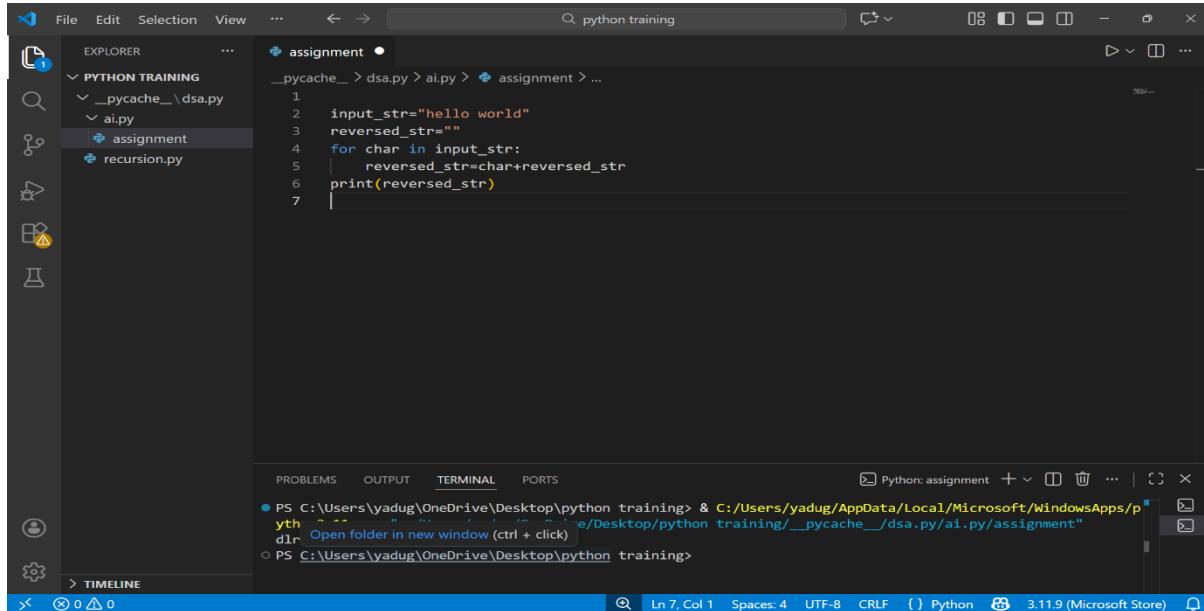


Assignment-1.5

Name: K.Likhith

RollNo:2303A51331

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



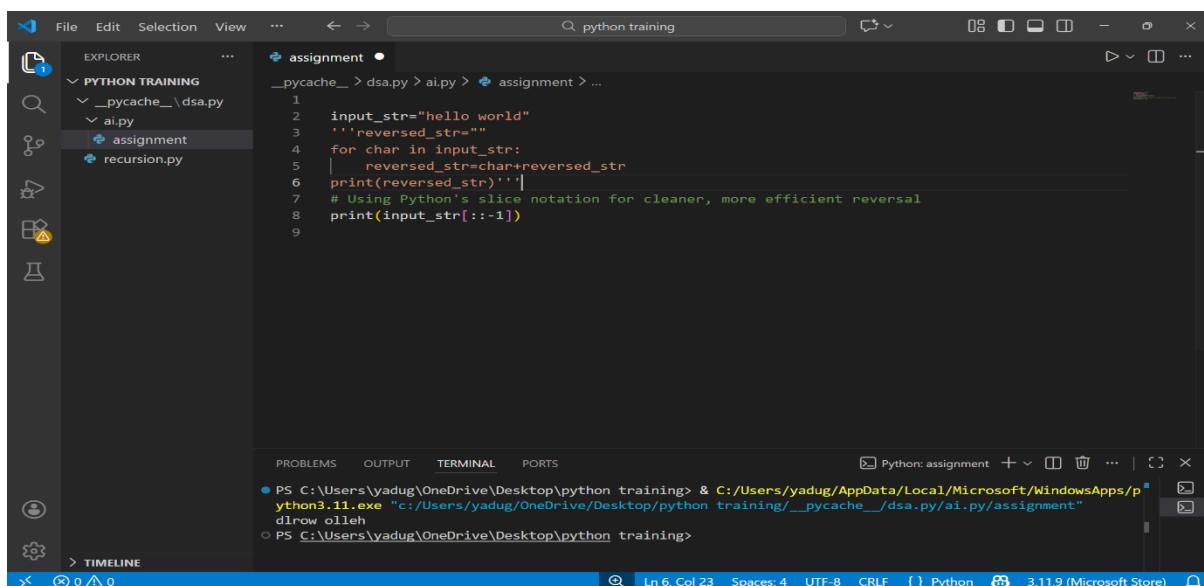
The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows a folder structure under "PYTHON TRAINING" containing "assignment" and "recursion.py".
- CODE EDITOR:** Displays the "assignment" file with the following Python code:

```
1 input_str="hello world"
2 reversed_str=""
3 for char in input_str:
4     reversed_str=char+reversed_str
5 print(reversed_str)
```
- TERMINAL:** Shows the command line output:

```
PS C:\Users\yadug\OneDrive\Desktop\python training> & C:/Users/yadug/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/yadug/OneDrive/Desktop/python training/_pycache__dsa.py/ai.py/assignment"
dlrow olleh
PS C:\Users\yadug\OneDrive\Desktop\python training>
```

TASK-2 Efficiency & Logic Optimization (Readability Improvement)



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

- EXPLORER:** Shows a folder structure under "PYTHON TRAINING" containing "assignment" and "recursion.py".
- CODE EDITOR:** Displays the "assignment" file with the following Python code:

```
1 input_str="hello world"
2 reversed_str=""
3 for char in input_str:
4     reversed_str=char+reversed_str
5 print(reversed_str)
6 # Using Python's slice notation for cleaner, more efficient reversal
7
8 print(input_str[::-1])
```
- TERMINAL:** Shows the command line output:

```
PS C:\Users\yadug\OneDrive\Desktop\python training> & C:/Users/yadug/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/yadug/OneDrive/Desktop/python training/_pycache__dsa.py/ai.py/assignment"
dlrow olleh
PS C:\Users\yadug\OneDrive\Desktop\python training>
```

- Explanation: `[::-1]` tells Python:
 - Start from end

- Move backwards
- Step = -1
- Python internally reverses the string in **one pass**.

Time Complexity

- Each character is accessed **once**.
- No repeated copying like the loop version.

Time Complexity: $O(n)$

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

Before optimization

```

FILE: assignment.py
1 def strreverse(s):
2     return s[::-1]
3 str1 = "Hello, World!"
4 print(strreverse(str1))
5 # Output: !dlroW ,olleH

TERMINAL OUTPUT:
PS C:\Users\yadug\OneDrive\Desktop\python training> & C:/Users/yadug/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/yadug/OneDrive/Desktop/python training/_pycache_/dsa.py/assignment"
PS C:\Users\yadug\OneDrive\Desktop\python training> & C:/Users/yadug/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/yadug/OneDrive/Desktop/python training/_pycache_/dsa.py/ai.py/assignment"
PS C:\Users\yadug\OneDrive\Desktop\python training>
  
```

After optimization using copilot

```

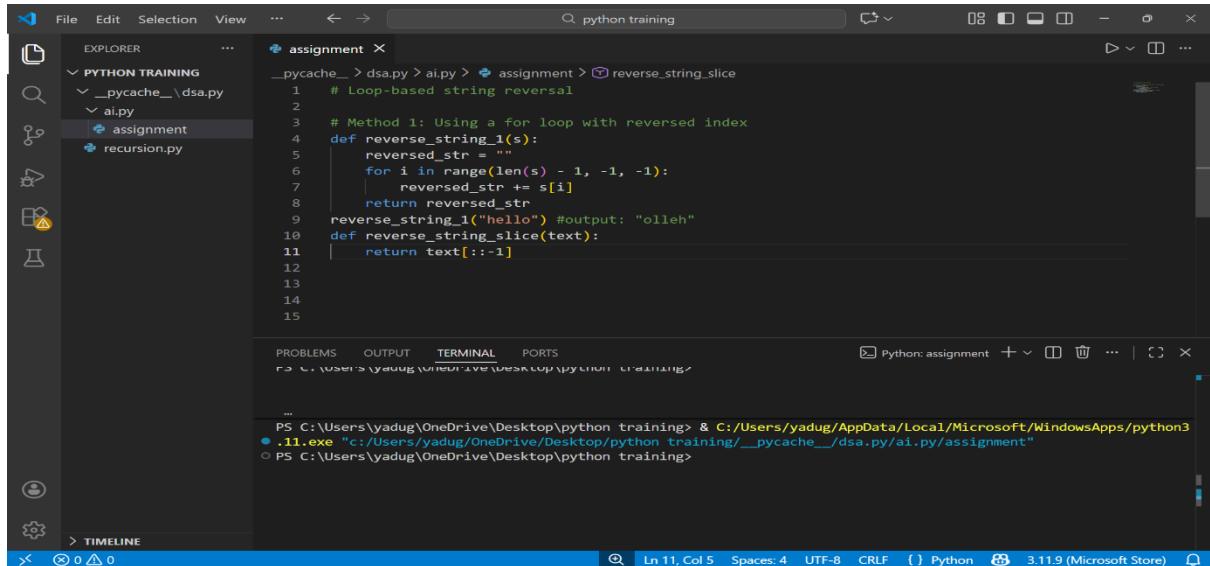
File Edit Selection View ... < > python training
EXPLORER PYTHON TRAINING _pycache_\dsa.py assignment ai.py recursion.py
assignment
assignment > strreverse(s):
    """Reverses the string.
    return s[::-1]
strreverse("hello") Example usage: returns "olleH"
PROBLEMS OUTPUT TERMINAL PORTS Python: assignment + < 1/3 > Accept Tab Accept Word Ctrl + RightArrow ...
PS C:\Users\yadug\OneDrive\Desktop\python training> & C:/Users/yadug/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/yadug/OneDrive/Desktop/python training/_pycache_\dsa.py/assignment"
PS C:\Users\yadug\OneDrive\Desktop\python training> & C:/Users/yadug/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/yadug/OneDrive/Desktop/python training/_pycache_\dsa.py/ai.py/assignment"
olleH
PS C:\Users\yadug\OneDrive\Desktop\python training>

```

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

Criteria	Without Functions (Procedural)	With Functions (Modular)
Code Clarity	Logic mixed with execution, harder to read in large files	Clear separation of logic and execution
Reusability	Code must be rewritten for every use	Function can be reused anywhere
Debugging Ease	Bugs are harder to isolate	Easy to debug by testing the function alone
Scalability	Poor for large programs	Highly suitable for large applications
Maintenance	Changes must be made in multiple places	Changes made once in the function
Testing	Manual testing only	Supports unit testing easily
Readability	Acceptable for small scripts	Excellent for professional codebases
Copilot Assistance	Limited suggestions	Strong AI support for docstrings & logic

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



```
File Edit Selection View ... ← → ⌂ python training ⌂ EXPLORER PYTHON TRAINING _pycache_\dsa.py ai.py assignment recursion.py

assignment
_pycache_ > dsa.py > ai.py > assignment > reverse_string_slice
1 # Loop-based string reversal
2
3 # Method 1: Using a for loop with reversed index
4 def reverse_string_1(s):
5     reversed_str = ""
6     for i in range(len(s) - 1, -1, -1):
7         reversed_str += s[i]
8     return reversed_str
9 reverse_string_1("hello") #output: "olleh"
10 def reverse_string_slice(text):
11     return text[::-1]

PROBLEMS OUTPUT TERMINAL PORTS Python: assignment + ⌂ ... x
PS C:\Users\yadug\OneDrive\Desktop\python training> & C:/Users/yadug/AppData/Local/Microsoft/WindowsApps/python3
.11.exe "c:/Users/yadug/OneDrive/Desktop/python training/_pycache_\dsa.py/ai.py/assignment"
PS C:\Users\yadug\OneDrive\Desktop\python training>

...
PS C:\Users\yadug\OneDrive\Desktop\python training> & C:/Users/yadug/AppData/Local/Microsoft/WindowsApps/python3
.11.exe "c:/Users/yadug/OneDrive/Desktop/python training/_pycache_\dsa.py/ai.py/assignment"
PS C:\Users\yadug\OneDrive\Desktop\python training>
```

Explanation:

Both approaches correctly reverse strings.

Slicing-based reversal is superior in performance, readability, and scalability.