

AI ASSISTANT CODING

Lab Assignment 1.5

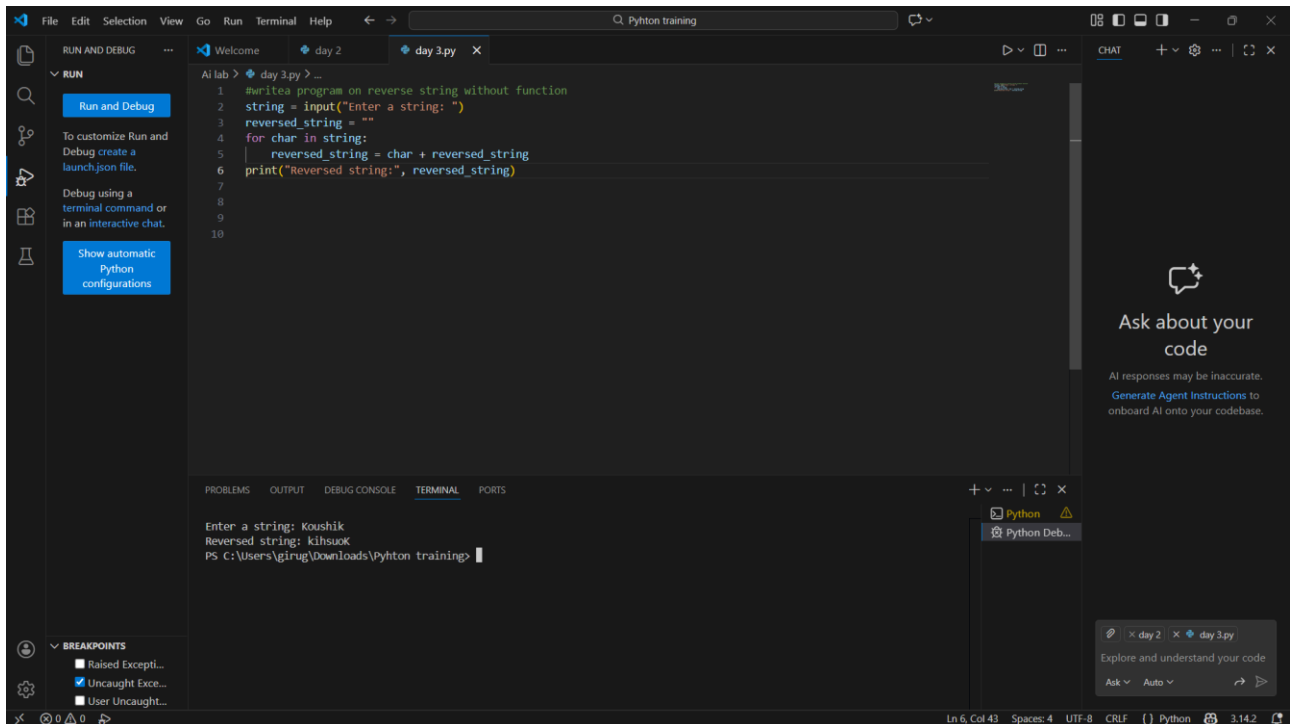
Name: GIRUGULA VARSHINI

2403A51L14

Batch : 51

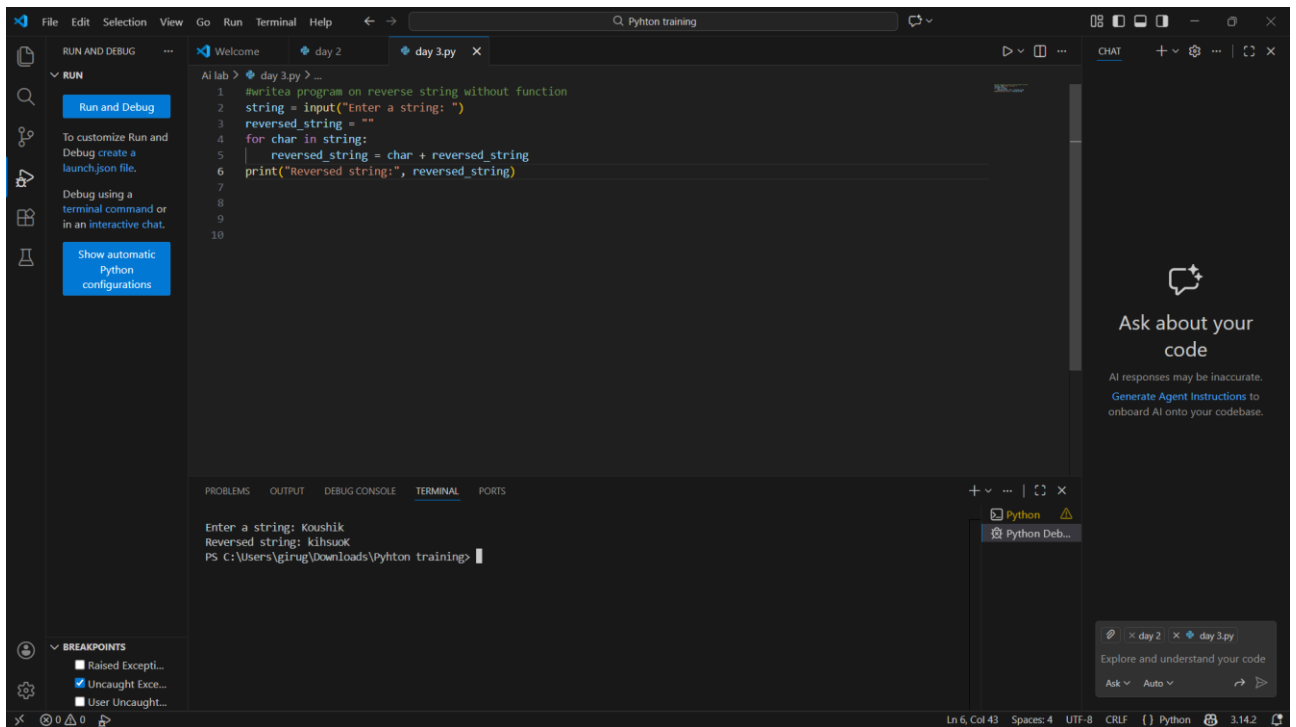
Task 1: AI-Generated Logic Without Modularization (Reverse String)

Prompt Used: “write a simple python program on Reverse String without using functions”

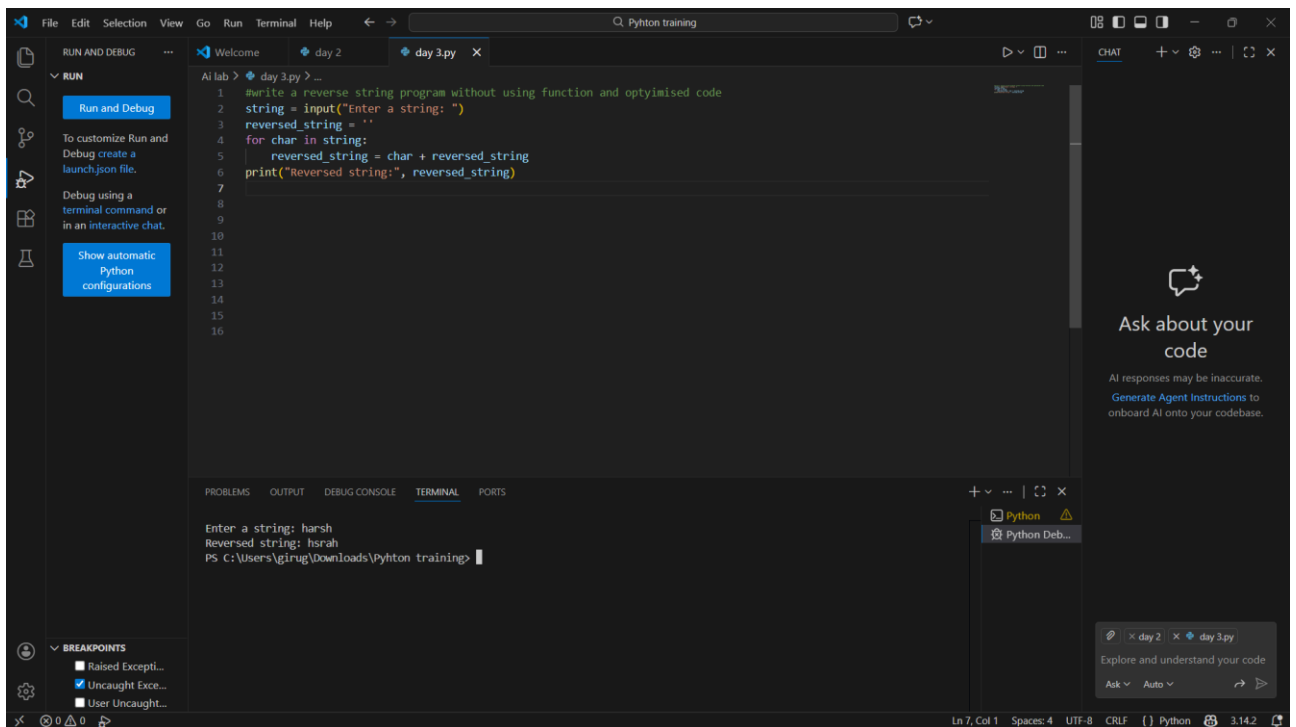


- Keeps the program simple
- Suitable for small scripts
- Easy for basic understanding
- No function call overhead

Task 2: AI Code Optimization & Cleanup Original Code:



Prompt Used: "optimize this code & simplify logic and improve readability"

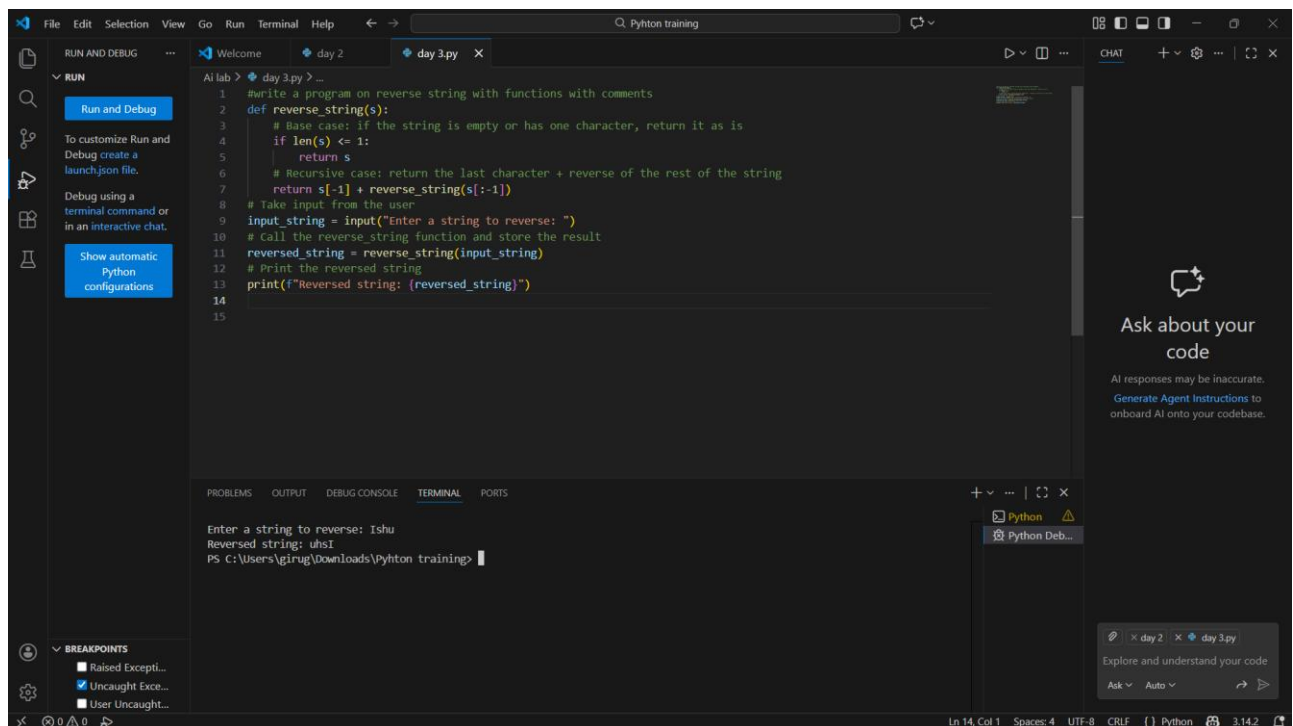


Code is cleaner and easier to maintain

The optimized version improves clarity, maintainability, and readability without affecting performance.

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

Prompt Used: “ Write a simple python program of using with function”



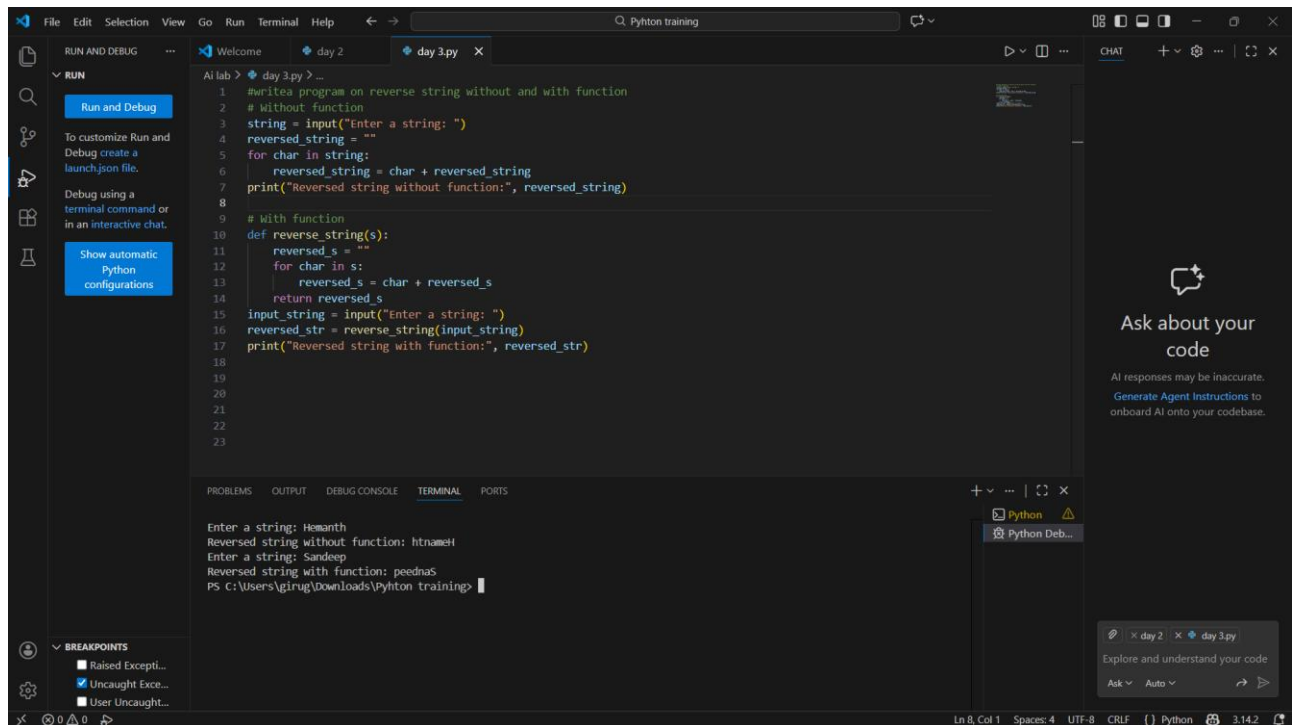
Using functions improves reusability because the same logic can be called multiple times.

It also improves readability and debugging.

Modular code is easier to maintain in large projects.

Task 4: With and Without Using Functions(Reverse String)

Prompt Used: “ Write a simple python program of Reverse String using with function and without using function”



The screenshot shows a Python IDE with a file named `day 3.py`. The code implements two methods to reverse a string: one without using a function and one using a function. The program prompts the user to enter a string and prints the reversed string for both methods. The terminal output shows the execution results for the input strings "Hemanth" and "Sandeep".

```
1 #write a program on reverse string without and with function
2 # Without function
3 string = input("Enter a string: ")
4 reversed_string = ""
5 for char in string:
6     reversed_string = char + reversed_string
7 print("Reversed string without function:", reversed_string)
8
9 # With function
10 def reverse_string(s):
11     reversed_s = ""
12     for char in s:
13         reversed_s = char + reversed_s
14     return reversed_s
15 input_string = input("Enter a string: ")
16 reversed_str = reverse_string(input_string)
17 print("Reversed string with function:", reversed_str)
18
19
20
21
22
23
```

Terminal Output:

```
Enter a string: Hemanth
Reversed string without function: htnameH
Enter a string: Sandeep
Reversed string with function: peeduaS
PS C:\Users\lgirug\Downloads\Python training>
```

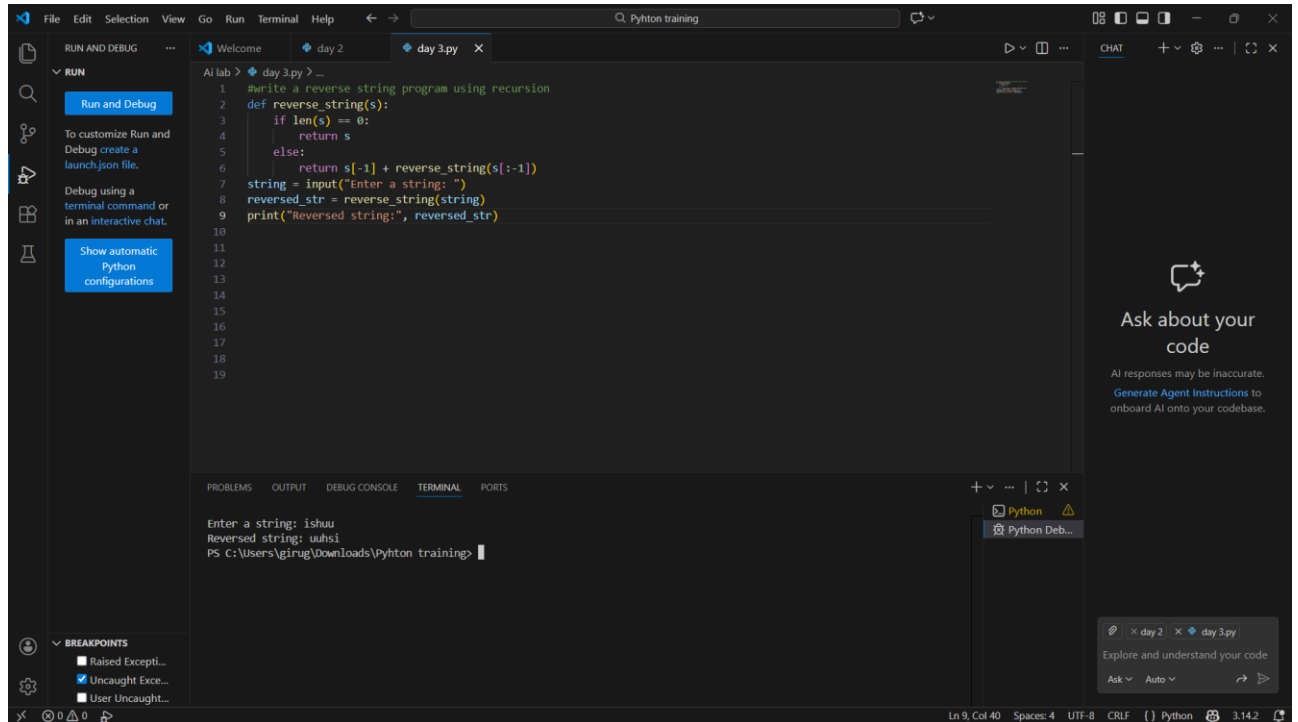
Without using functions: Helps beginners clearly understand the basic logic and step-by-step execution of an Armstrong number program.

Using functions: Makes the code modular, reusable, and easier to read and maintain.

Overall: Using functions follows good programming practices, especially for larger or real-world programs.

Task 5: Iterative vs Recursive AI Code

Prompt Used: “Generate iterative and recursive Reverse String in Python”



The screenshot shows a Python IDE with a file named `day 3.py` open. The code implements a recursive function `reverse_string(s)` to reverse a string. The function uses a base case where if the length of the string is 0, it returns the string. Otherwise, it returns the last character of the string concatenated with the reverse of the substring excluding the last character. The main program prompts the user to enter a string, calls the `reverse_string` function, and prints the result.

```
1 #write a reverse string program using recursion
2 def reverse_string(s):
3     if len(s) == 0:
4         return s
5     else:
6         return s[-1] + reverse_string(s[:-1])
7 string = input("Enter a string: ")
8 reversed_str = reverse_string(string)
9 print("Reversed string:", reversed_str)
```

The terminal output shows the execution of the program:

```
Enter a string: ishuv
Reversed string: uhsui
PS C:\Users\girug\Downloads\Pyhton training>
```

The IDE interface includes a sidebar with 'RUN AND DEBUG' options, a 'CHAT' panel on the right with a prompt 'Ask about your code', and a status bar at the bottom showing file information and line/col counts.

Execution Flow Explanation

- Iterative version uses loops
- Recursive version uses function calls
- Recursive calls stack memory