

# AI ASSISTANT CODING

## Lab Assignment 1.5

Name: A.Abhiram

2403A51L13

Batch : 51

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

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

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

- File Menu:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Toolbar:** RUN AND DEBUG, RUN, CHAT.
- Code Editor:** Shows a Python file named "day 3.py" with the following code:

```
AI lab > #write a program on reverse string without function
1 string = input("Enter a string: ")
2 reversed_string = ""
3 for char in string:
4     |     reversed_string = char + reversed_string
5 print("Reversed string:", reversed_string)
```
- Terminal:** Shows the output of running the script: "Enter a string: Koushik" followed by "Reversed string: khsuok".
- Breakpoints:** A list of breakpoints including "Raised Excepti...", "Uncought Excepti...", and "User Uncought...".
- Bottom Status Bar:** Shows file information: Ln 6, Col 43, Spaces: 4, CRLF, Python, 3.14.2.

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

## Task 2: AI Code Optimization & Cleanup Original Code:

A screenshot of the Visual Studio Code interface. The top bar shows 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', 'Terminal', 'Help'. The title bar says 'Python training'. The main editor window contains the following Python code:

```
1 #write a program on reverse string without function
2 string = input("Enter a string: ")
3 reversed_string = ""
4 for char in string:
5     reversed_string = char + reversed_string
6 print("Reversed string:", reversed_string)
```

The 'RUN' section of the sidebar has 'Run and Debug' selected. The terminal below shows the output of running the code with the input 'Koushik':

```
Enter a string: Koushik
Reversed string: khsuok
PS C:\Users\girug\Downloads\Python training>
```

The bottom right corner of the interface features an AI agent icon with the text 'Ask about your code'.

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

A screenshot of the Visual Studio Code interface, identical to the previous one but with optimized code. The main editor window now contains:

```
1 #write a reverse string program without using function and optymised code
2 string = input("Enter a string: ")
3 reversed_string = ...
4 for char in string:
5     reversed_string = char + reversed_string
6 print("Reversed string:", reversed_string)
```

The terminal output remains the same as the first screenshot.

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”

A screenshot of the Visual Studio Code (VS Code) interface. The top bar shows the file path "Ai lab > day 3.py". The main editor area contains the following Python code:

```
#write a program on reverse string with functions with comments
def reverse_string(s):
    # Base Case: if the string is empty or has one character, return it as is
    if len(s) <= 1:
        return s
    # Recursive case: return the last character + reverse of the rest of the string
    return s[-1] + reverse_string(s[:-1])
# Take input from the user
input_string = input("Enter a string to reverse: ")
# Call the reverse_string function and store the result
reversed_string = reverse_string(input_string)
# Print the reversed string
print(f"Reversed string: {reversed_string}")
```

The sidebar on the left includes a "RUN AND DEBUG" section with a "Run and Debug" button highlighted. Below it are sections for "To customize Run and Debug create a launch.json file.", "Debug using a terminal command or in an interactive chat.", and "Show automatic Python configurations".

The bottom right corner of the editor has a "CHAT" tab with a message bubble icon and the text "Ask about your code". Below this, a note says "AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase."

The bottom status bar shows the terminal output: "Enter a string to reverse: Ishu Reversed string: uhsI PS C:\Users\girug\Downloads\Python training>".

The bottom right corner also shows the file status: "day 3.py" is the active file. The status bar at the bottom indicates "Ln 14, Col 1" and "Spaces: 4" and "UTF-8".

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 the Visual Studio Code interface with two tabs open: 'day 2' and 'day 3.py'. The 'day 3.py' tab contains the following code:

```
AI lab > day 3.py > ...
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
```

The terminal below shows the output of running the script:

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

The left sidebar shows the 'RUN AND DEBUG' section with 'Run and Debug' selected. The bottom right corner has a 'CHAT' button and a message 'Ask about your code'.

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 the Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder structure with 'RUN AND DEBUG' and 'RUN' sections.
- Code Editor:** Displays a Python script named 'day 3.py' containing the following code:

```
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)
```
- Terminal:** Shows the output of running the script: "Enter a string: ishuu" followed by "Reversed string: uuhsi".
- Bottom Status Bar:** Shows file paths ('day 2', 'day 3.py'), line count (Ln 9, Col 40), spaces (Spaces: 4), encoding (UTF-8), and other settings.

### Execution Flow Explanation

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