

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”

The screenshot shows a Python code editor interface. On the left, there's a sidebar with 'RUN AND DEBUG' options, including 'Run and Debug' which is highlighted. Below it are instructions to 'Customize Run and Debug' and 'Show automatic Python configurations'. The main area displays a Python script named 'day 3.py' with the following code:

```
Al 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)
```

Below the code, the terminal window shows the output of running the script with the input 'Koushik':

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

The bottom right corner features an AI interface with a message bubble icon and the text 'Ask about your code'. It also includes a note: 'AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase.'

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

Task 2: AI Code Optimization & Cleanup Original Code:

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

- File Explorer:** Shows a folder structure with 'RUN AND DEBUG' and 'day 3.py'.
- Code Editor:** Displays 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)
```
- Terminal:** Shows the output of running the code: "Enter a string: Koushik\nReversed string: kihsuok".
- Output:** Shows a 'Python' output channel.
- Problems:** No errors or warnings are listed.
- Breakpoints:** A breakpoint is set at line 3.
- Status Bar:** Shows 'Ln 6, Col 43' and other file-related information.

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

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

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

```
1 #write a reverse string program without using function and optyimised 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)
```
- Terminal:** Shows the output of running the code: "Enter a string: harsh\nReversed string: hsrah".
- Output:** Shows a 'Python' output channel.
- Problems:** No errors or warnings are listed.
- Breakpoints:** A breakpoint is set at line 3.
- Status Bar:** Shows 'Ln 7, Col 1' and other file-related information.

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 window displays the following Python code:

```
1 #write a program to reverse string with functions with comments
2 def reverse_string(s):
3     # Base case: if the string is empty or has one character, return it as is
4     if len(s) <= 1:
5         return s
6     # Recursive case: return the last character + reverse of the rest of the string
7     return s[-1] + reverse_string(s[:-1])
8 # Take input from the user
9 input_string = input("Enter a string to reverse: ")
10 # Call the reverse_string function and store the result
11 reversed_string = reverse_string(input_string)
12 # Print the reversed string
13 print(f"Reversed string: {reversed_string}")
14
15
```

To the right of the code editor, there is an AI assistance panel 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 of the screen shows the terminal output:

```
Enter a string to reverse: Isha
Reversed string: hsIa
PS C:\Users\girug\Downloads\Python training>
```

The bottom status bar indicates "Ln 14, Col 1 Spaces: 4 UTF-8 CRLF { } Python 3.14.2".

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 Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows a folder named "RUN AND DEBUG".
- Run and Debug:** A button with the text "Run and Debug".
- Terminal:** The title bar says "Python training". The terminal window displays the output of running the script "day 3.py".

```
Enter a string: Hemant
Reversed string without function: tnameH
Enter a string: Sandeep
Reversed string with function: peednas
PS C:\Users\girug\Downloads\Python training>
```
- Breakpoints:** A sidebar showing breakpoints for the current file.
- Code Editor:** The main editor area contains the following Python code:

```
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)
```
- AI Assistant:** A floating message bubble with the text "Ask about your code".
- Bottom Status Bar:** Shows line 8, column 1; spaces 4; UTF-8; CRLF; Python; and line numbers 1-23.

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 (VS Code) interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar says "Python training". The left sidebar has sections for RUN AND DEBUG, RUN, and BREAKPOINTS. The main area displays a Python script named "day 3.py" with 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)
```

The terminal below shows the output of running the script:

```
Enter a string: ishuu
Reversed string: uuhsi
PS C:\Users\girug\Downloads\Python training>
```

The status bar at the bottom indicates "Ln 9, Col 40" and "3.14.2".

Execution Flow Explanation

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