

NAME =>Sai akshith

BATCH =>45

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

❖ Scenario

You are developing a basic text-processing utility for a messaging application.

CODE:

```
# generate a code for a string reversal without using functions  
input_string = "Hello, World!"  
reversed_string = ""  
for char in input_string:  
    reversed_string = char + reversed_string  
print("Reversed string:", reversed_string)
```

The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING> & C:/Python314/python.exe "c:/Users/SAIAKSHITH/OneDrive/Documents/AI-PROGRAMMING/exp 1.py"
Reversed string: !dlrow ,olleH
PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING>
```

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

❖ Scenario

The string reversal logic is needed in multiple parts of an application.

CODE:

```
def reverse_string(s):
    reversed_s = ""
    for char in s:
        reversed_s = char + reversed_s
    return reversed_s

input_string = "Hello, World!"

print("Reversed string using function:", reverse_string(input_string))
```

The screenshot shows a code editor interface with multiple tabs at the top: 'Welcome', 'exp4-3.py', 'exp 5.py', 'exp 6-2.py', 'exp 1.py', and 'exp 1.py > ...'. The active tab is 'exp 1.py > ...'. The code in the editor is:

```
8 # generate a code for a string reversal using functions
9 def reverse_string(s):
10     reversed_s = ""
11     for char in s:
12         reversed_s = char + reversed_s
13     return reversed_s
14 input_string = "Hello, World!"
15 print("Reversed string using function:", reverse_string(input_string))
```

Below the code editor is a terminal window with the following output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING> & C:/Python314/python.exe "c:/Users/SAIAKSHITH/OneDrive/Documents/AI-PROGRAMMING/exp 1.py"
Reversed string using function: !dlroW ,olleH
PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING> []
```

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

❖ Scenario

Your mentor wants to evaluate how AI handles alternative logic paths

CODE:

```
# generate a code for reversal of a string using slicing and loop

def reverse_string_slicing(s):

    return s[::-1]

input_string = "Hello, World!"

print("Reversed string using slicing:", reverse_string_slicing(input_string))

def reverse_string_loop(s):

    reversed_s = ""

    for char in s:

        reversed_s = char + reversed_s
```

```
return reversed_s

input_string = "Hello, World!"

print("Reversed string using loop:", reverse_string_loop(input_string))
```

```
16 # generate a code for reversal of a string using slicing and loop
17 def reverse_string_slicing(s):
18     return s[::-1]
19 input_string = "Hello, World!"
20 print("Reversed string using slicing:", reverse_string_slicing(input_string))
21 def reverse_string_loop(s):
22     reversed_s = ""
23     for char in s:
24         reversed_s = char + reversed_s
25     return reversed_s
26 input_string = "Hello, World!"
27 print("Reversed string using loop:", reverse_string_loop(input_string))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING> & C:/Python314/python.exe "c:/Users/SAIAKSHITH/OneDrive/Documents/AI-PROGRAMMING/exp1.py"
Reversed string using function: !dlrow ,olleH
Reversed string using slicing: !dlrow ,olleH
Reversed string using loop: !dlrow ,olleH
PS C:\Users\SAIAKSHITH\OneDrive\Documents\AI-PROGRAMMING> []
```