

AI ASSISTED CODING

LAB-1.5

Katta Lasya

2303A51724

Batch-11

Task 0

- ❖ Install and configure GitHub Copilot in VS Code. Take screenshots of each step.

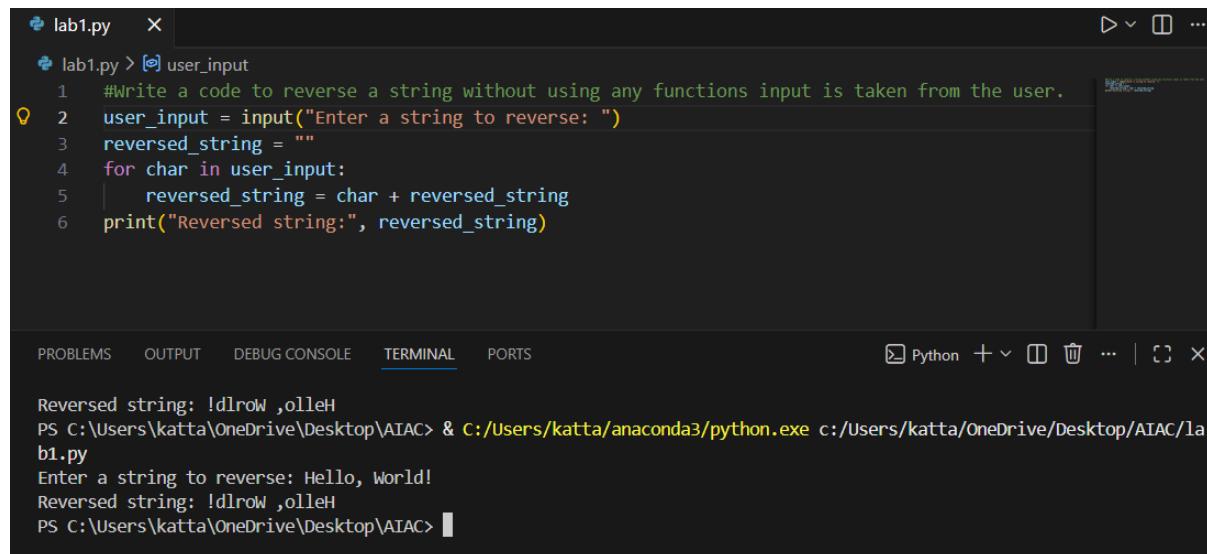
Already did it in previous semesters.

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

Prompt:

Write a code to reverse a string without using any functions, input is taken from the user.

Given Code and Output:



The screenshot shows a VS Code interface with a dark theme. A code editor window titled 'lab1.py' contains the following Python code:

```
#Write a code to reverse a string without using any functions input is taken from the user.
user_input = input("Enter a string to reverse: ")
reversed_string = ""
for char in user_input:
    reversed_string = char + reversed_string
print("Reversed string:", reversed_string)
```

Below the code editor is a terminal window showing the execution of the script and its output:

```
Reversed string: !dlrow ,olleH
PS C:\Users\katta\OneDrive\Desktop\AIAC> & C:/Users/katta/anaconda3/python.exe c:/Users/katta/OneDrive/Desktop/AIAC/lab1.py
Enter a string to reverse: Hello, World!
Reversed string: !dlrow ,olleH
PS C:\Users\katta\OneDrive\Desktop\AIAC>
```

Explanation:

- At first I just wrote “Write a code to reverse a string” then it gave the code using functions.
- Then again I wrote “Write a code to reverse a string without using any functions” then it gave the code without using any functions but the input is given in the code and not taken by the user.

- Then after giving the prompt “Write a code to reverse a string without using any functions, input is taken from the user.” It generated the code that satisfied the requirements in the question.

Task 2: Efficiency & Logic Optimization (Readability Improvement)

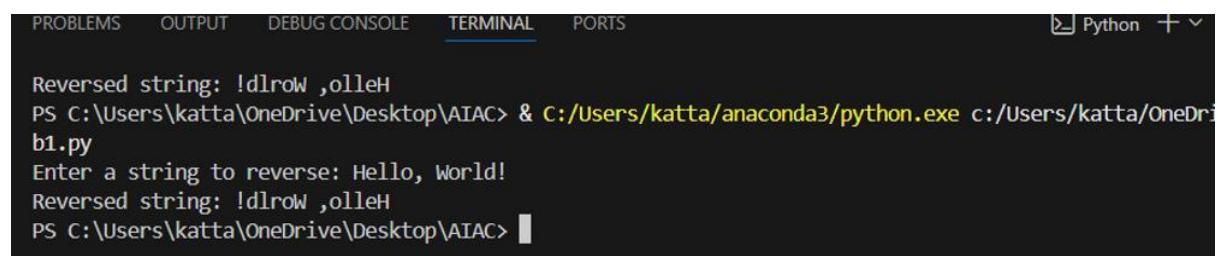
Prompt:

Optimize the code below by adding comments and improving readability.

Code:

```
lab1.py > ...
1  #Write a code to reverse a string without using any functions input is taken from the user.
2  #Optimize the code below by adding comments and improving readability.
3  # Prompt the user to enter a string
4  user_input = input("Enter a string to reverse: ")
5  # Initialize an empty string to store the reversed string
6  reversed_string = ""
7  # Iterate through each character in the input string
8  for char in user_input:
9      # Prepend the current character to the reversed string
10     reversed_string = char + reversed_string
11 # Display the reversed string to the user
12 print("Reversed string:", reversed_string)
13 # This code reverses a string input by the user without using any built-in functions.
```

Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + v
Reversed string: !dlrow ,olleH
PS C:\Users\katta\OneDrive\Desktop\AIAC> & C:/Users/katta/anaconda3/python.exe c:/Users/katta/OneDr
b1.py
Enter a string to reverse: Hello, World!
Reversed string: !dlrow ,olleH
PS C:\Users\katta\OneDrive\Desktop\AIAC>
```

Explanation:

- I wrote “Optimize the code below by removing the redundant variable and simplifying the logic” but as the below attached image it did not give any response because the code is already simplified.

```
lab1.py > ...
1  #Write a code to reverse a string without using any functions input is taken from the user.
2  #Optimize the code below by removing the redundant variable and simplifying the logic.
3  |
4  user_input = input("Enter a string to reverse: ")
5  reversed_string = ""
6  for char in user_input:
7      reversed_string = char + reversed_string
8  print("Reversed string:", reversed_string)
9
10
```

- So, gave the prompt as “Optimize the code below by adding comments and improving readability.” Then the response was shown by explaining every line of the code to easily understand it.

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

Prompt:

```
def reverse_string_input_user():
```

Code and Output:

```
lab1.py > ...
1 # def reverse_string_input_user():
2 def reverse_string_input_user():
3     user_input = input("Enter a string: ")
4     reversed_string = user_input[::-1]
5     print("Reversed string:", reversed_string)
6     return reversed_string
7 #function call
8 reverse_string_input_user()
9
```

Explanation:

- By giving the prompt as “def reverse_string_input_user():”
 - The response is given with the function name same as the function name used for the prompt
 - Just using words to specify the code that is required is simple but is a little tricky
 - It took me 3 to 4 tries to get the code that I wanted with this def function method but it’s easy without writing some sentences and will get a hang of it after practicing some.

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

Procedural Approach (Without Function)

- The entire code for reversing a string is written directly without using any function.
 - The logic is written step by step in one place.
 - It is easy to understand for small programs.
 - However, the code becomes lengthy and cannot be reused.
 - Making changes or debugging is a little difficult.

Modular Approach (With Function)

- By giving the prompt as “def reverse_string_input_user():”
 - The response is given using the same function name.

- Only simple words are used to specify what the code should do.
 - This method is simple but a little tricky at first.
 - It took a few tries to get the correct code using the function.
 - After practicing, it becomes easy and saves time.
 - The function can be reused whenever needed.

Task 5: AI-Generated Iterative vs Recursive Fibonacci Approaches (Different)

Algorithmic Approaches to String Reversal

Using loops:

Write a code to reverse a given string with loops without using functions input by the user.

Code and Output:

```
lab1.py > ...
1 # Write a code to reverse a given string with loops without using functions input by the user.
2 user_input = input("Enter a string to reverse: ")
3 reversed_string = ""
4 for char in user_input:
5     reversed_string = char + reversed_string
6 print("Reversed string:", reversed_string)
```

Explanation:

- By giving the prompt the response given is the code without any functions.
 - It used `reversed_string = " "` to store the reversed string
 - It gave the code with a for loop which also increases some time complexity.
 - It prints the reversed string at the end.

Using Slicing:

Prompt:

Write a code to reverse a given string with inbuilt function input by user.

Code and Output:



```
# Write a code to reverse a given string with inbuilt function input by user
user_input = input("Enter a string to reverse: ")
reversed_string = user_input[::-1]
print("Reversed string:", reversed_string)
```

Explanation:

- When the prompt is given as “Write a code to reverse a given string with inbuilt function input by user.”
 - The response code is given with using slicing for the reversing
 - It is stored in reversed_string and it is printed.