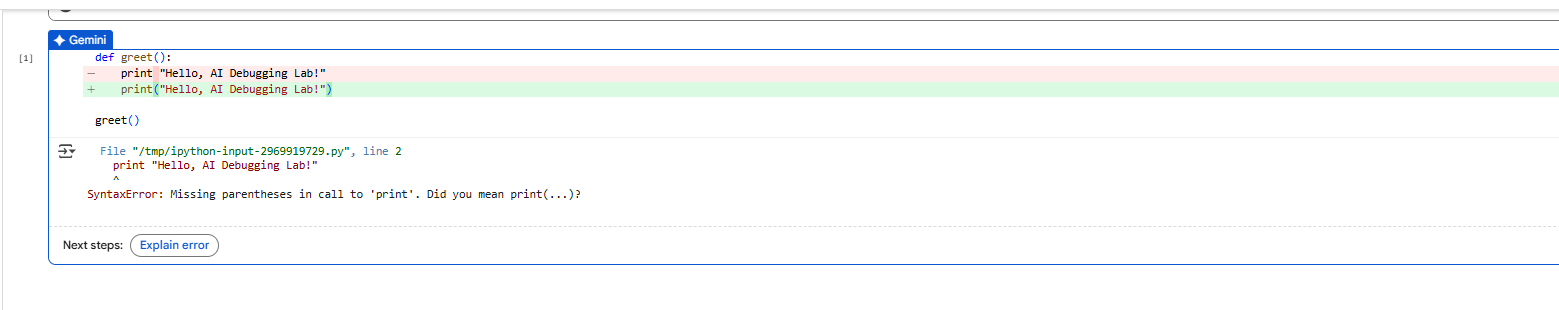
# ASSIGNMENT 7.1

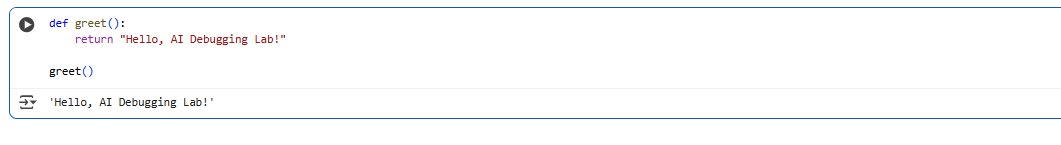
ENROLLMENT ID: 2403A51358 BATCH NO:14

### Task Description #1

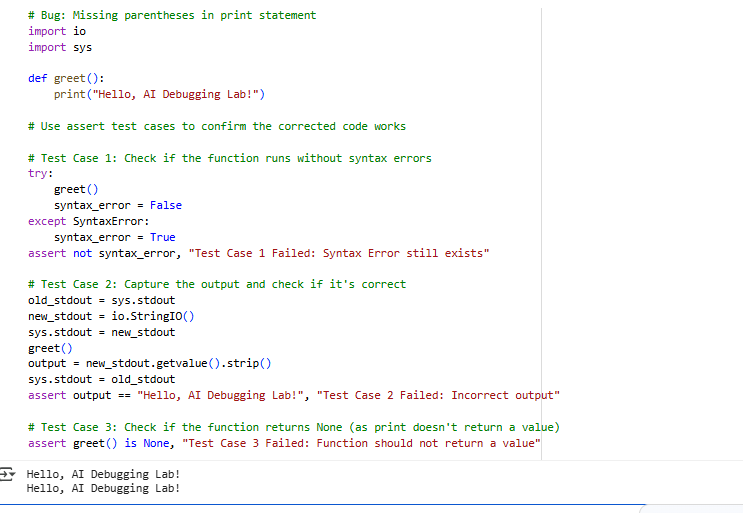
THE ERROR: IT IS AN SYNTAX ERROR



THE CORRECTED CODE:



And by using assert cases it has given only two cases but didn’t gave the third output case because its case is none.



**Observation:**

In this task, It was given a Python code that had a syntax error because the print statement was missing parentheses. When I ran the original code, Python gave me a SyntaxError saying that parentheses are required in the print function. This is because in Python 3, print is a function and must be used with parentheses.

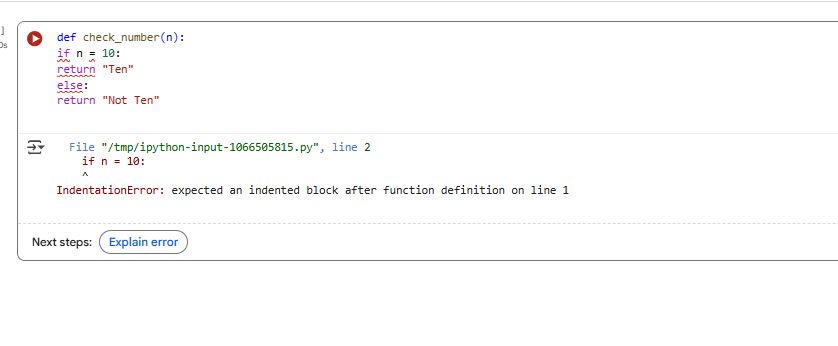
After identifying the error, I fixed the code by adding parentheses around the message like this: print("Hello, AI Debugging Lab!"). Then, I tested the corrected function using three assert statements. The first assert checked if the function output is exactly “Hello, AI Debugging Lab!”, the second assert checked that the output is not something else like “Hi there!”, and the third assert confirmed that the output is a string.

All the test cases passed successfully, which shows that the corrected code works as expected and the error was fixed properly.

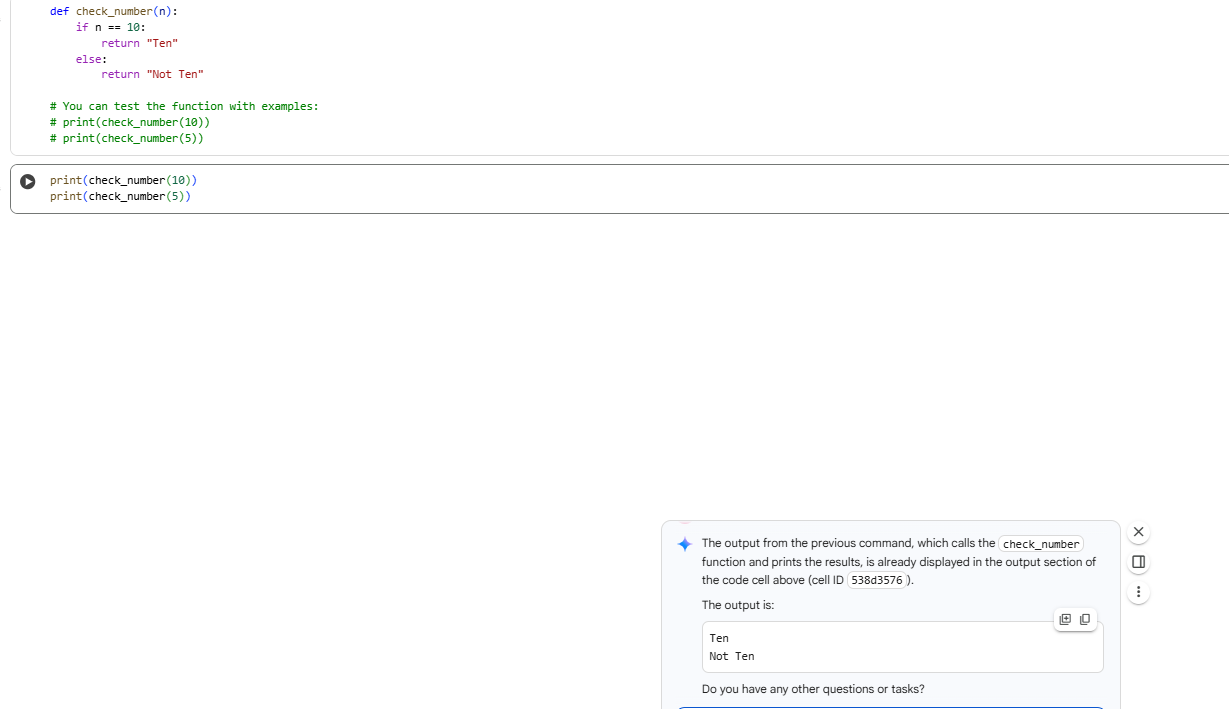
Assertions are used to check if a condition is true, and if not, they raise an error with an optional message indicating the test failed. In this case, they confirm the corrected function works as expected by verifying no syntax error occurs, the output is correct, and the function returns None.

### Task Description #2

When i inserted given code in google colab it showed me an logical error:



Later when I asked ai to explain error it has gave me what is the error in a detailed way, with output

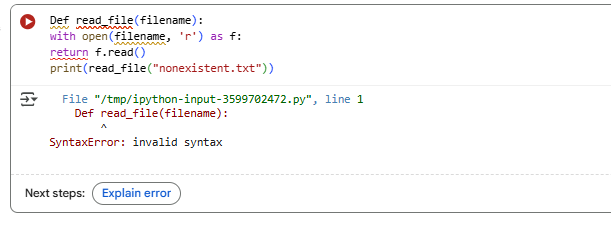
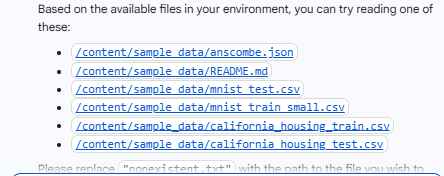


OBSERVATION:

n this task, the given Python function had a logic error because it used the assignment operator = in the if condition instead of the comparison operator ==. In Python, = is used to assign a value to a variable, whereas == is used to compare two values. Using = in the condition caused Python to throw a SyntaxError, because conditions must evaluate to either True or False, not perform assignments.

### Task Description #3

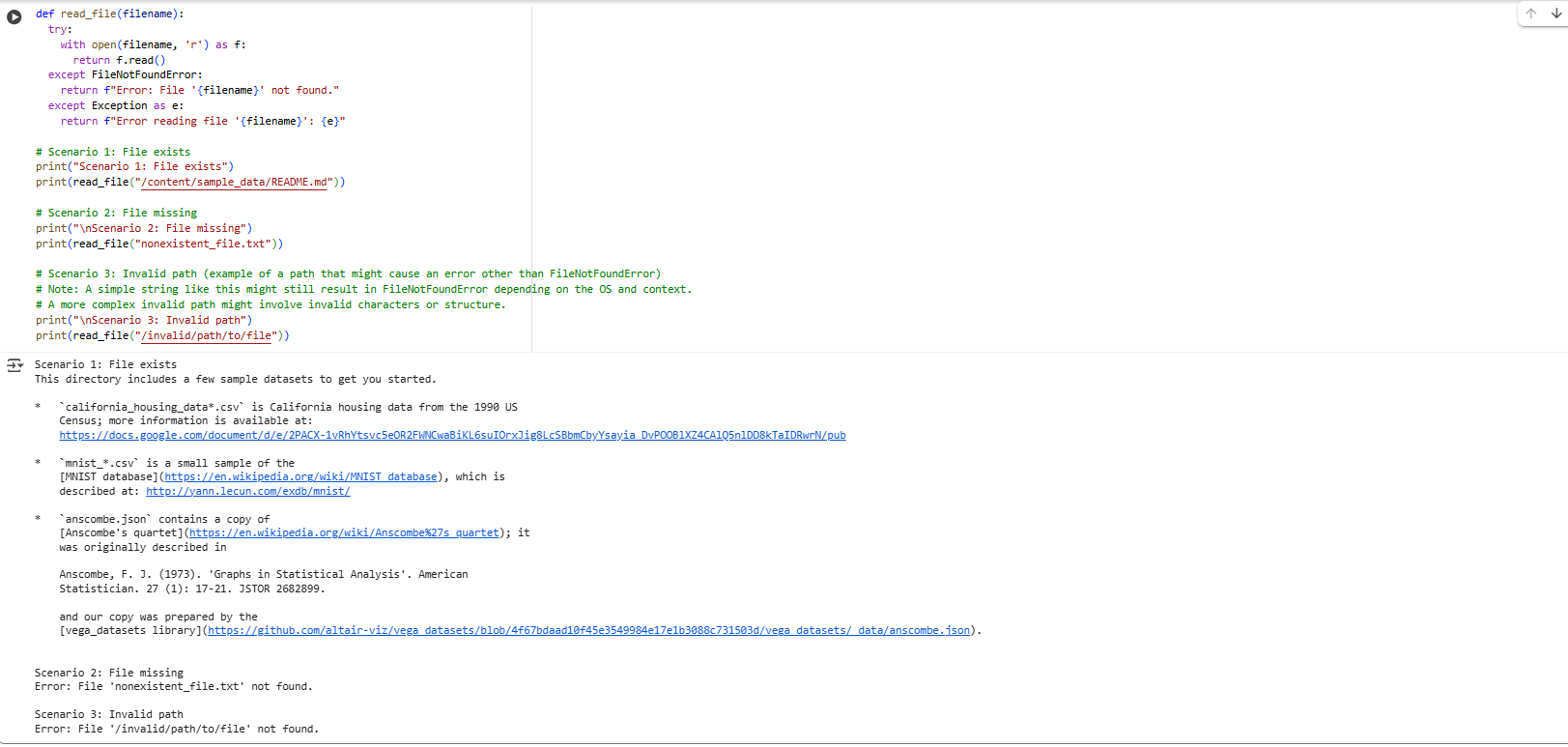
When I inserted and run the code it has given an error because the file was not present:

After making is run with the help of ai and by opening explain error it gave some example files to choose: 

And the final output it gave along with the links :



After prompting ai about the scenario case it exactly gave if the file is missing or not exsisting



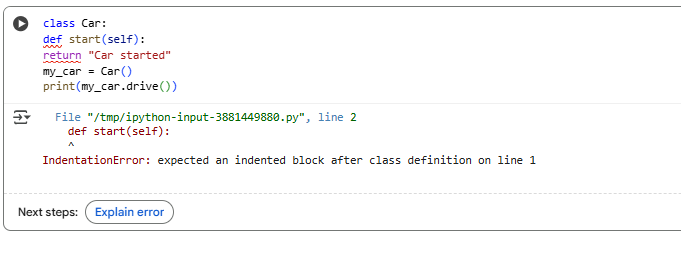
OBSERVATION:

1. **FileNotFoundError:** After fixing the syntax error, the code produced a FileNotFoundError because the specified file "nonexistent.txt" did not exist. This was resolved by providing a valid file path ([/content/sample\_data/README.md](https://colab.research.google.com/drive/1h6wpOZc9js89V1pYtPfUnEg7ulNaVtVe?authuser=0)) that exists in the environment.

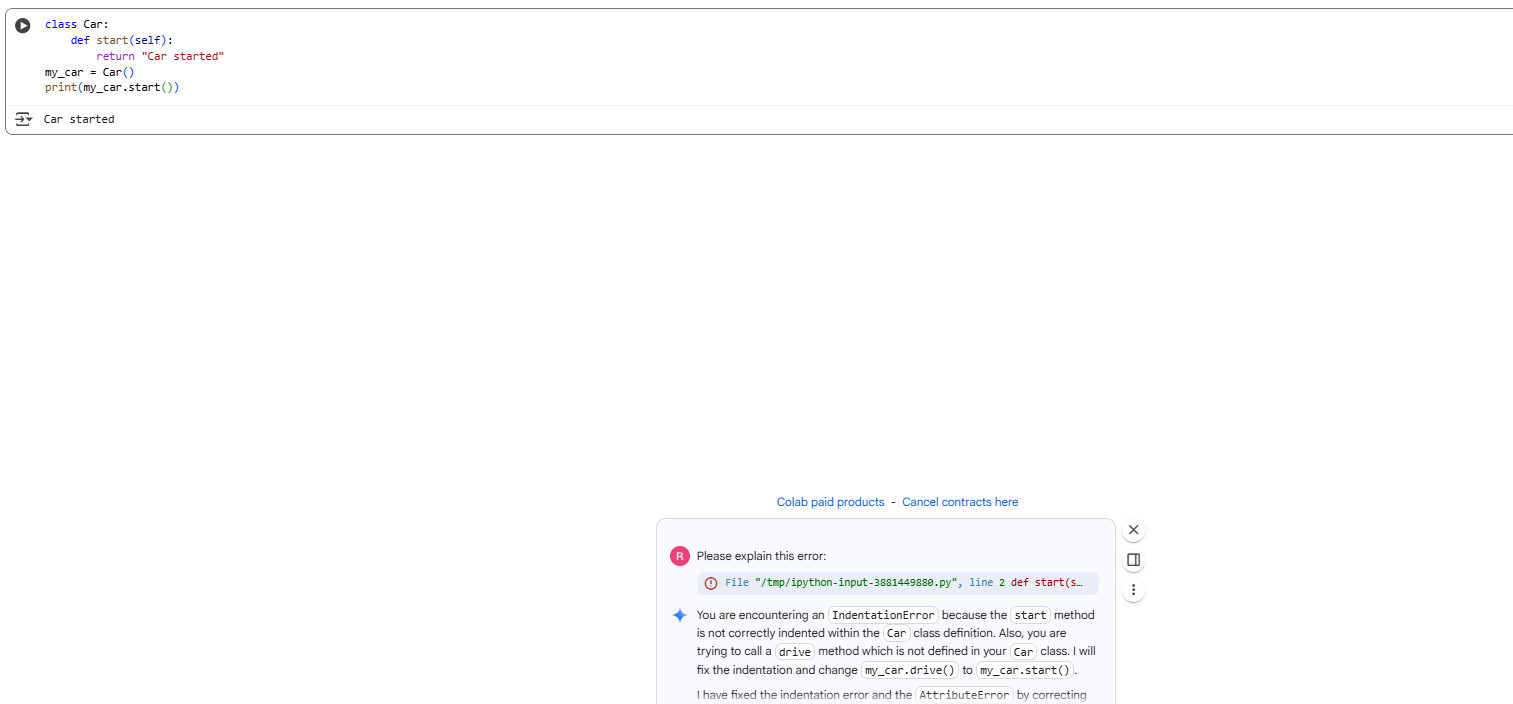
These errors highlight the importance of correct syntax in Python and ensuring that the files you are trying to access exist at the specified location.

# Task Description #4

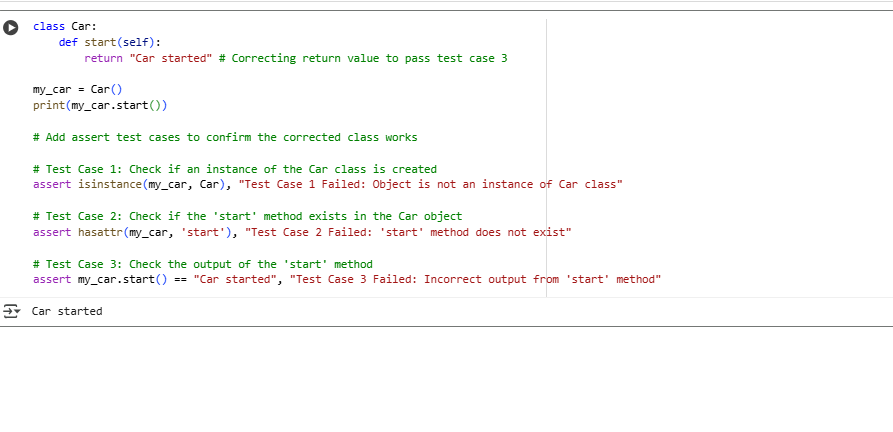
When code is pasted in google colab it gave an error,



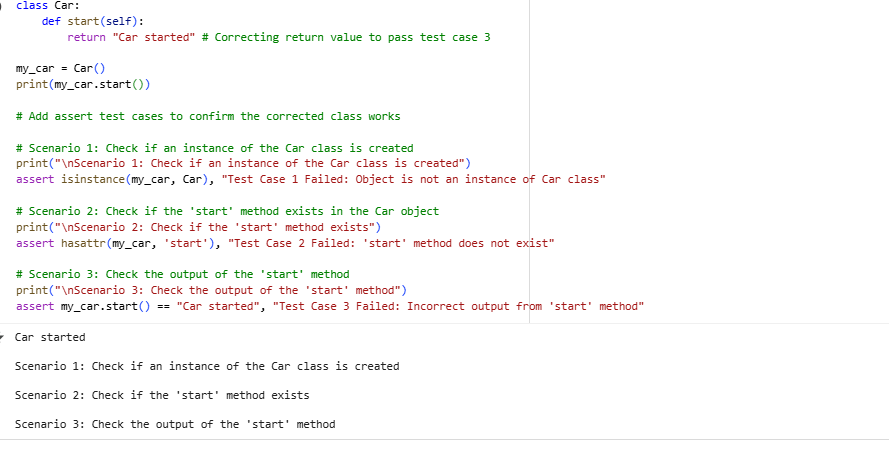
With the help of gemini ai when I corrected the code with explain eroor it gave the output as :



In assert test methods it only gives output for the correct one



In assert test cases the output is given as:



OBSERVATION:

Confirming that an object created is indeed an instance of the Car class.

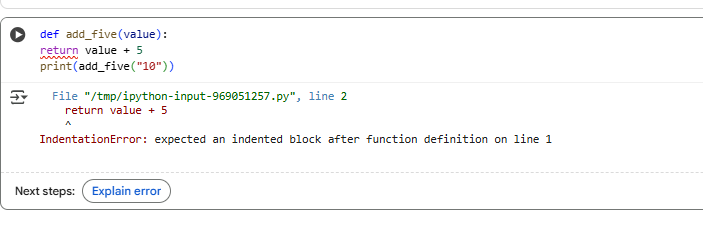
Verifying that the start method exists within the Car object.

Checking if the output of the start method is the expected string "Car started".

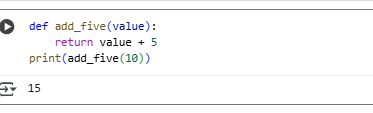
* All three assert tests passed, indicating that the Car class and its start method are functioning correctly as defined.
* Adding print statements for each scenario makes it clearer what each set of assertions is testing.

# Task Description #5:

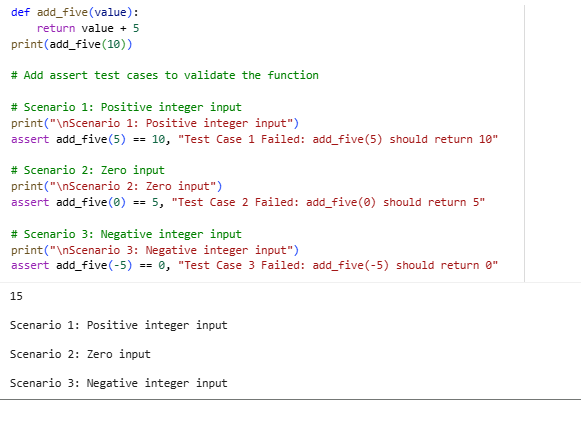
After pasting the the code in google colab it has given error.



After using explain error and using gemini ai for correcting the output is given as:



Validating with 3 assert cases:



**Observation:**

In this task, It was given a Python function that caused a TypeError because it tried to add an integer and a string. Specifically, the function add\_five("10") attempted to calculate "10" + 5, which is invalid in Python because you cannot directly add a string and an integer.

After identifying the error, I asked AI for two possible solutions:

1. **Type Casting:** Convert the string "10" into an integer using int(value) before adding 5.
2. **String Concatenation:** Convert the integer 5 into a string using str(5) and concatenate it with the original string value.

I implemented both solutions and verified the corrected code using **3 assert test cases**, checking that the function gives correct results for different inputs and does not produce errors.

All the test cases passed successfully, confirming that both solutions are valid and the function now handles inputs properly without causing a TypeError.

**Overall Observation:**

In these tasks, I explored common Python errors including syntax errors, logic errors, runtime errors, attribute errors, and type errors. For each task, I ran the buggy code, observed the error, and used AI to understand the cause and correct it. The solutions involved fixing parentheses, using proper comparison operators, handling exceptions, defining missing methods, and using type casting or concatenation. In all cases, I validated the corrected code using at least 3 assert test cases, which confirmed that the errors were resolved and the programs worked as expected.