**ASSIGNMENT-7.1**

**Name:P.Navya**

**HTNO:2403A51331**

**Batch:13**

**Prompt:**

(Syntax Errors – Missing Parentheses in Print  
Statement)  
Provide a Python snippet with a missing parenthesis in a print  
statement (e.g., print "Hello"). Use AI to detect and fix the syntax error.  
# Bug: Missing parentheses in print statement

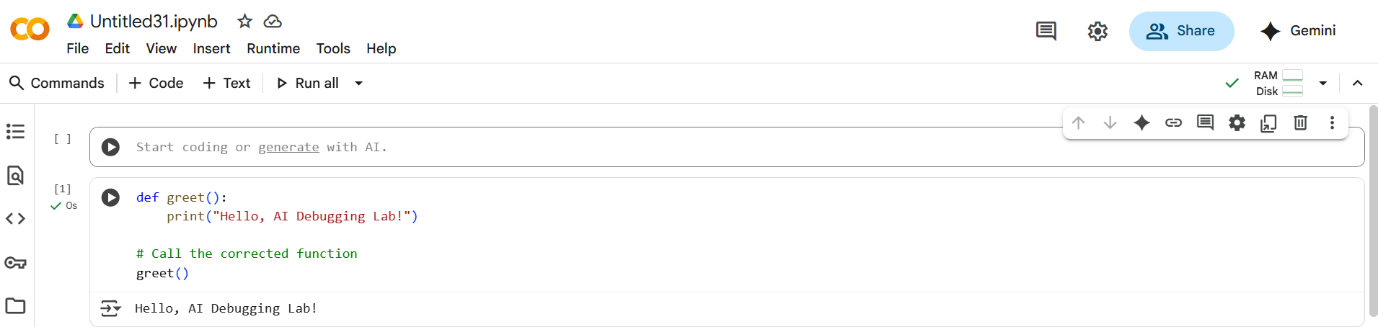
Missing Parentheses:

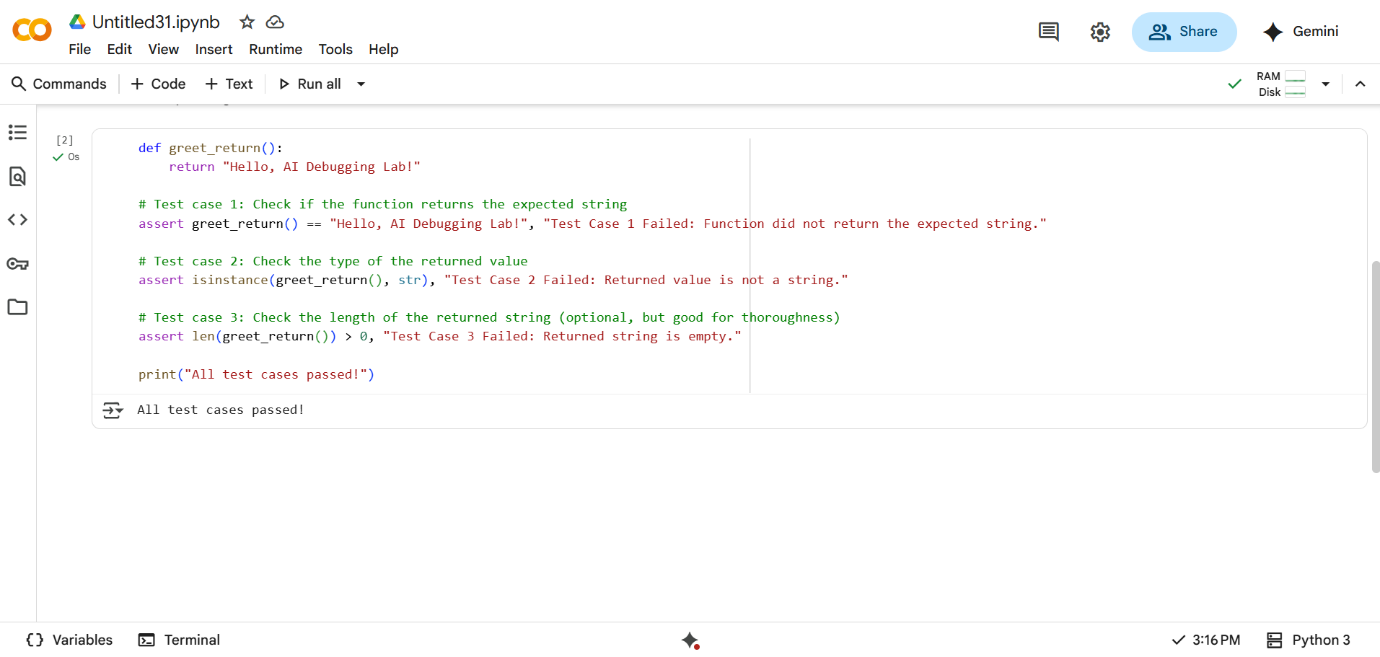
def greet():  
 print "Hello, AI Debugging Lab!"  
greet()

**Explanation:**

The original code had a SyntaxError because the print statement in Python 3 requires parentheses around the argument. The corrected code adds these parentheses**.**

corrected code:





**Task-2:** (Logic Error – Incorrect Condition in an If  
Statement)

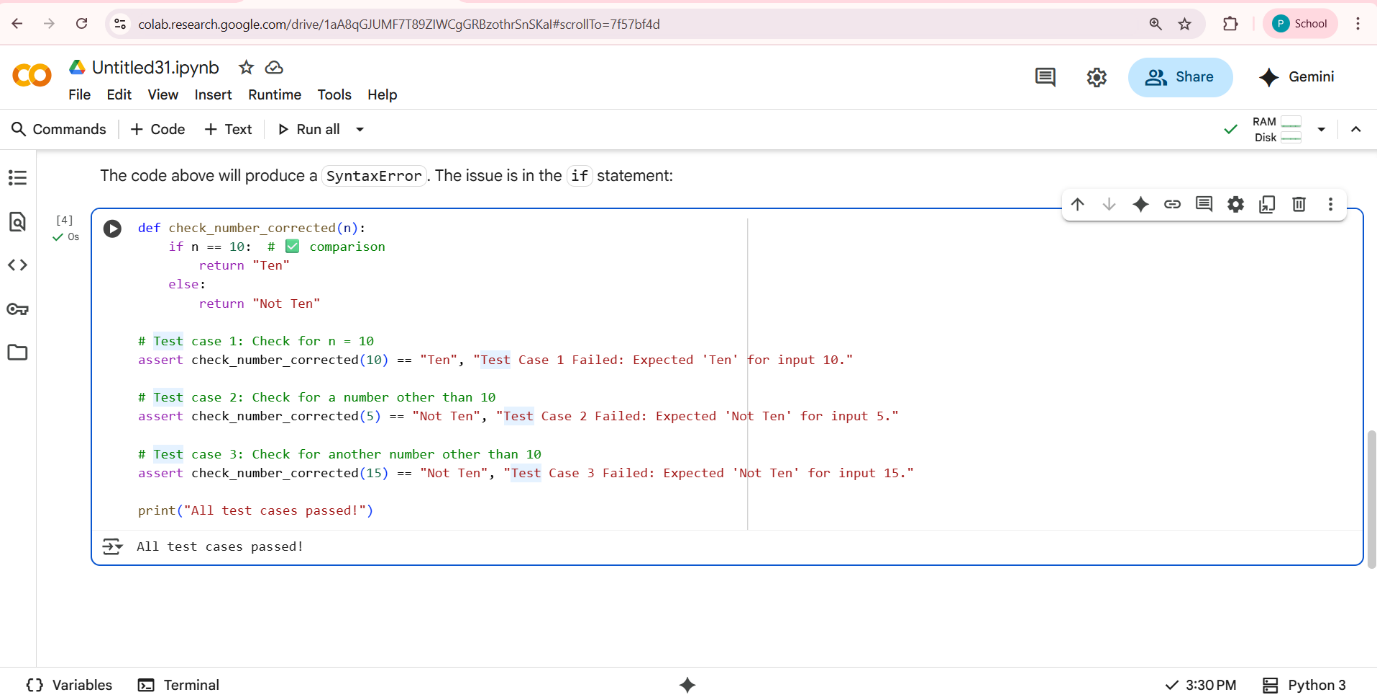
**Prompt**: Supply a function where an if-condition mistakenly uses = instead  
of ==. Let AI identify and fix the issue.  
# Bug: Using assignment (=) instead of comparison (==)



**Observation:**

The code above will produce a SyntaxError. The issue is in the if statement

**Corrected code:**

****

**Task-3:**

**(Runtime Error – File Not Found)**

**Prompt:**Provide code that attempts to open a non-existent file and crashes.  
Use AI to apply safe error handling.  
# Bug: Program crashes if file is missing  
def read\_file(filename):  
with open(filename, 'r') as f:  
return f.read()  
print(read\_file("nonexistent.txt"))

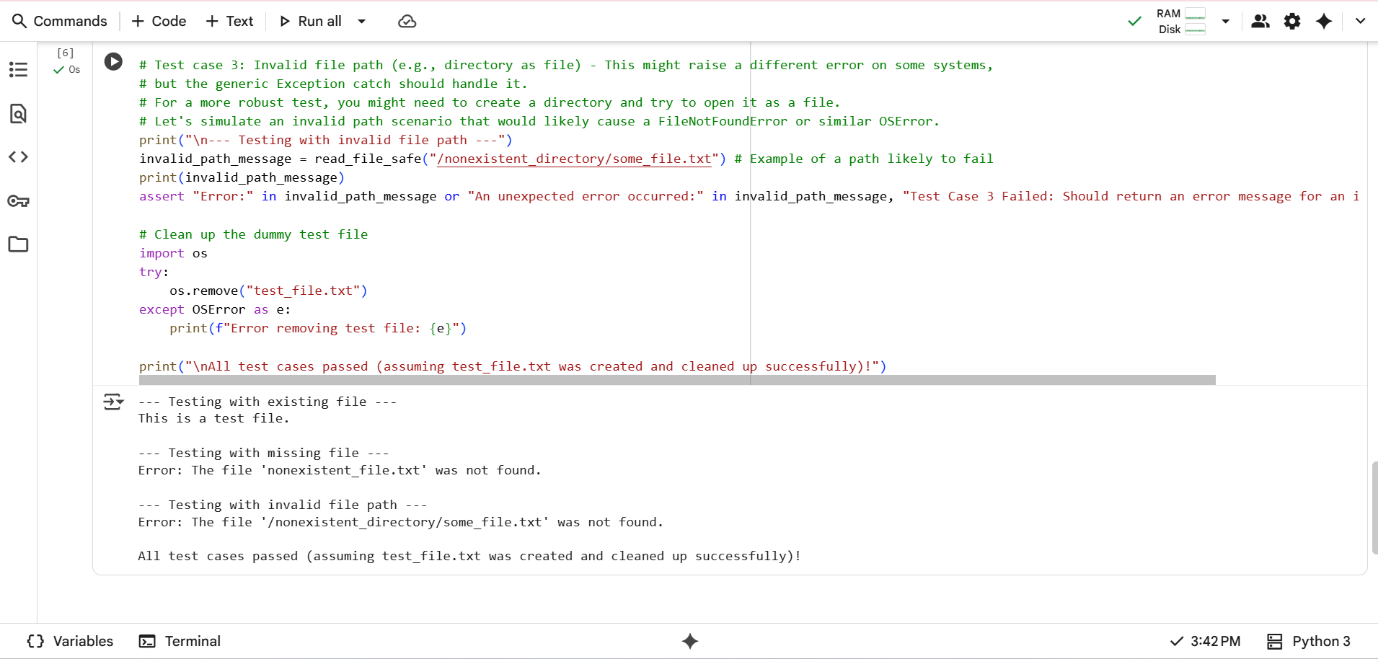


**Observation:**

running the code above results in a FileNotFoundError because the file "nonexistent.txt" does not exist. To handle this gracefully, we can use a try-except block.

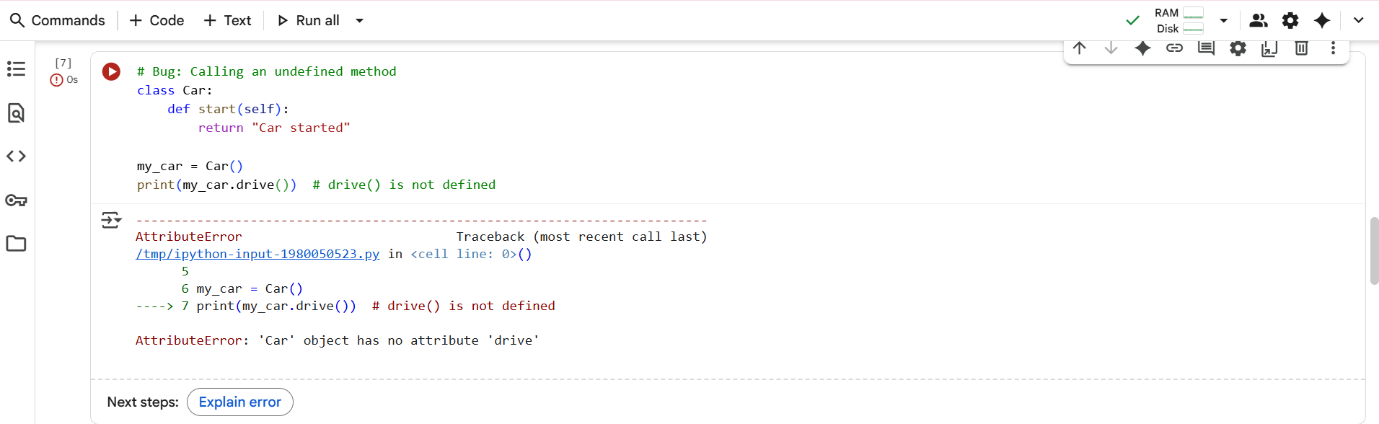
**Corrected code:**

****

****

**Task-4:**

(AttributeError – Calling a Non-Existent Method)

**Prompt:** Give a class where a non-existent method is called (e.g.,  
obj.undefined\_method()). Use AI to debug and fix.  
# Bug: Calling an undefined method  


**Observation:**

Running the code above will result in an AttributeError because you are trying to call a method named drive() on the Car object, but no such method is defined within the Car class.

To fix this, you have two main options:

1. **Define the missing method:** If you intended the Car class to have a drive method, you need to add a definition for it within the class.
2. **Correct the method call:** If you intended to call an existing method (like start()), you need to change the method call to match the defined method name.

Based on the existing start method, let's assume you intended to call start(). We will correct the method call.

**Corrected code:**

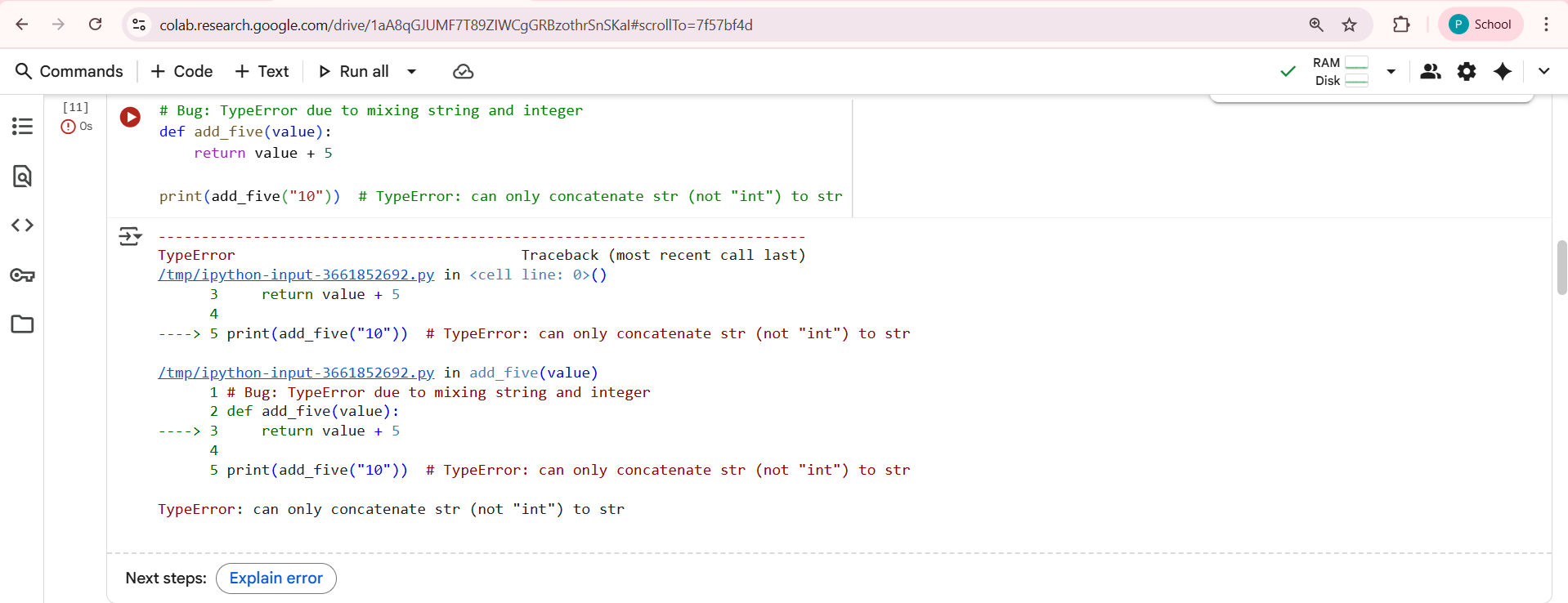
****

**Task-5:**

**(TypeError – Mixing Strings and Integers in  
Addition)**

**Prompt:**Task: Provide code that adds an integer and string ("5" + 2) causing a  
TypeError. Use AI to resolve the bug.

# Bug: TypeError due to mixing string and integer



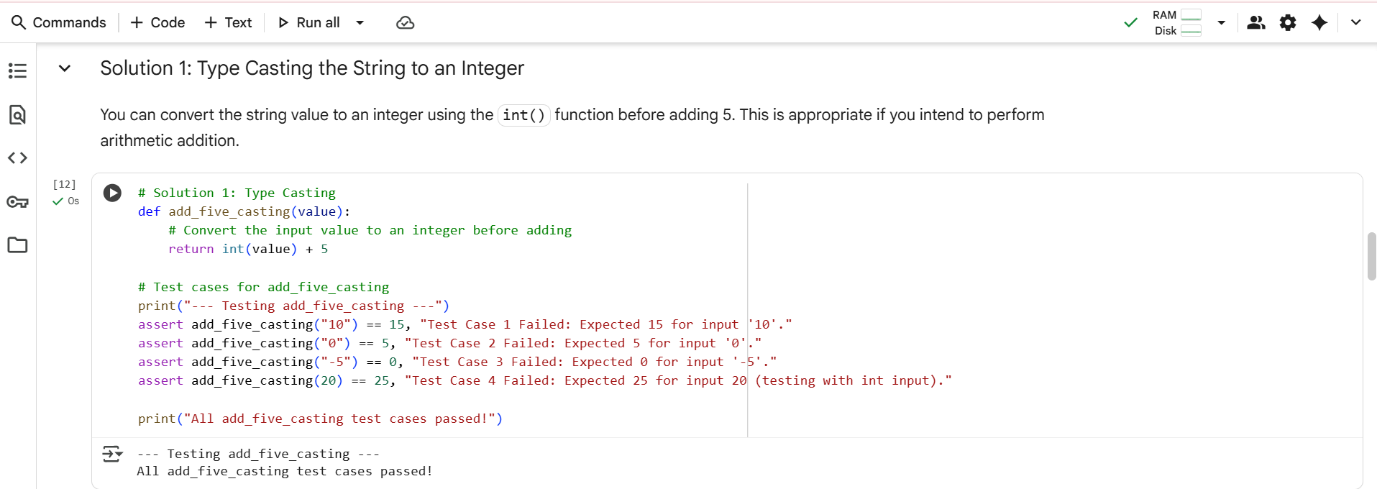
Observation:

Running the code above will result in a TypeError. This happens because you are trying to use the + operator to combine a string ("10") and an integer (5). The + operator has different meanings depending on the data types it's used with:

* For numbers, + performs arithmetic addition.
* For strings, + performs concatenation (joining strings together).

Python doesn't automatically know whether you want to do arithmetic or concatenation when you mix these types, so it raises a TypeError.

Here are two common approaches:





**Observation:**

You can convert the string value to an integer using the int() function before adding 5. This is appropriate if you intend to perform arithmetic addition.

You can convert the string value to an integer using the int() function before adding 5. This is appropriate if you intend to perform arithmetic addition.