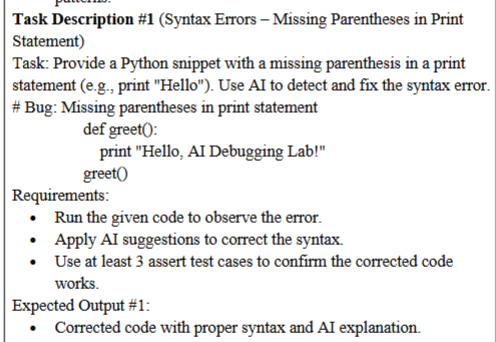
AI -ASSISTED-CODING

ASSIGNMENT-7.1

2403A51252

Batch-11

Task-1 :



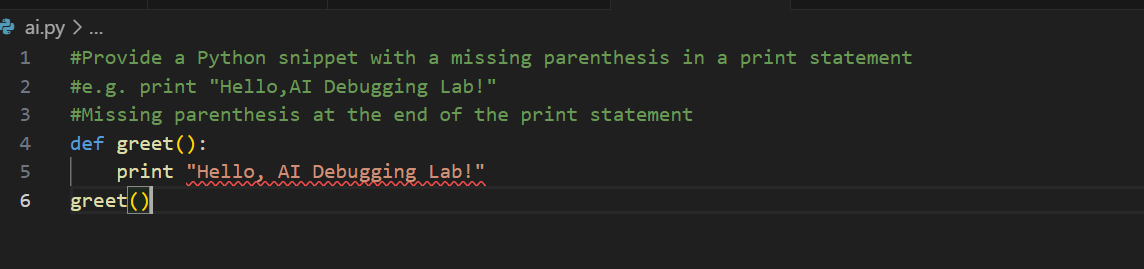
PROMPT :

#Provide a Python snippet with a missing parenthesis in a print statement

#e.g. print "Hello,AI Debugging Lab!"

#Bug: Missing parenthesis at the end of the print statement

CODE:



Explanation:

ERROR :

🡪Missing paranthesis in print statement

* print "Hello,AI Debugging Lab!"

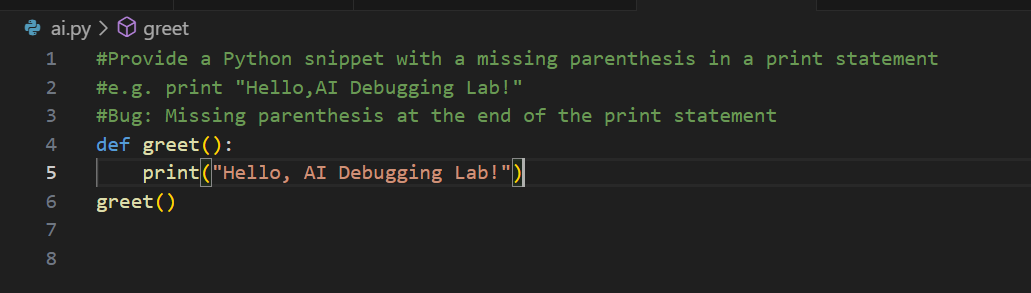
CORRECTION :

🡪Replaced the old-style print with Python 3 style print(...).

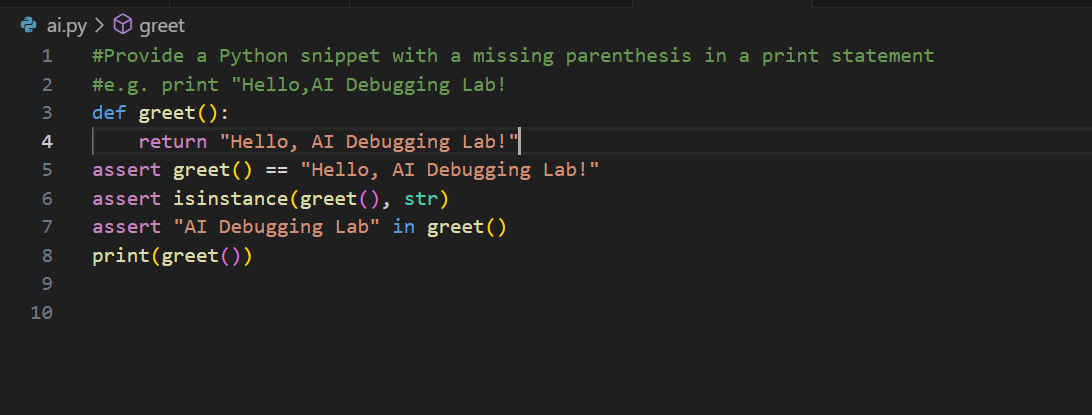
🡪Returned the greeting string from the function so we can test it using assert.

🡪Added assert statements to ensure the function behaves as expected

Corrected code:

1. 

2.



OUTPUT:

Hello,AI Debugging Lab!

TASK-2:

Task: Supply a function where an if-condition mistakenly uses = instead of ==. Let AI identify and fix the issue.

# Bug: Using assignment (=) instead of comparison (==)

def check\_number(n):

if n = 10:

return "Ten"

else:

return "Not Ten"

Requirements:

• Ask AI to explain why this causes a bug.

• Correct the code and verify with 3 assert test cases.

Expected Output #2:

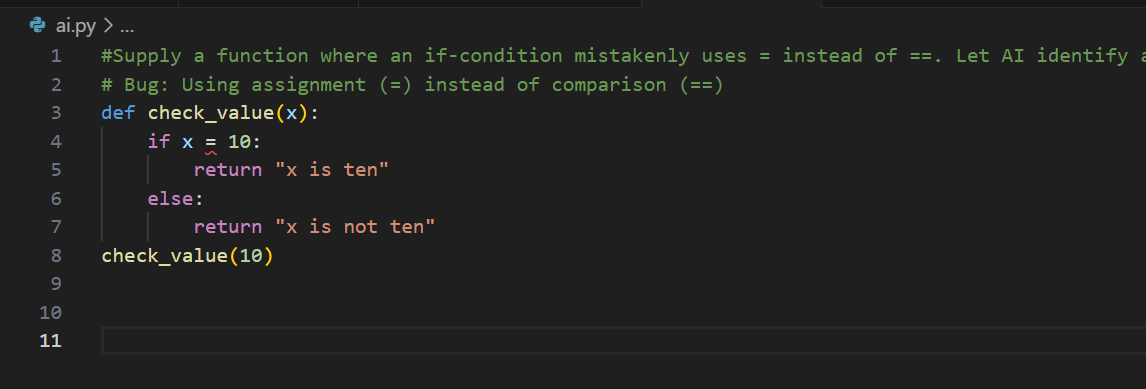
• Corrected code using == with explanation and successful test execution

PROMPT:

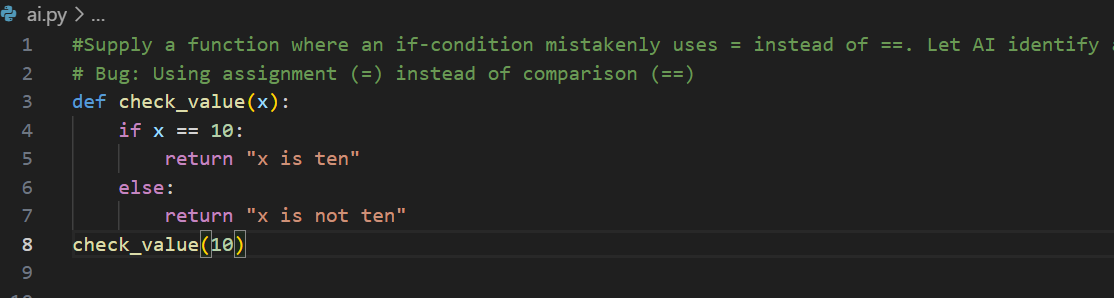
#Supply a function where an if-condition mistakenly uses = instead of ==. Let AI identify and fix the issue.

# Bug: Using assignment (=) instead of comparison (==)

CODE:



Corrected code:



Explantion:

* In Python, the single equals sign = is used for assignment, not comparison.
* In an if statement, Python expects a boolean expression, like n == 10.
* Writing if n = 10: tries to assign 10 to n inside the if, which is not allowed and results in a syntax error.

OUTPUT:

10 is ten

TASK-3:

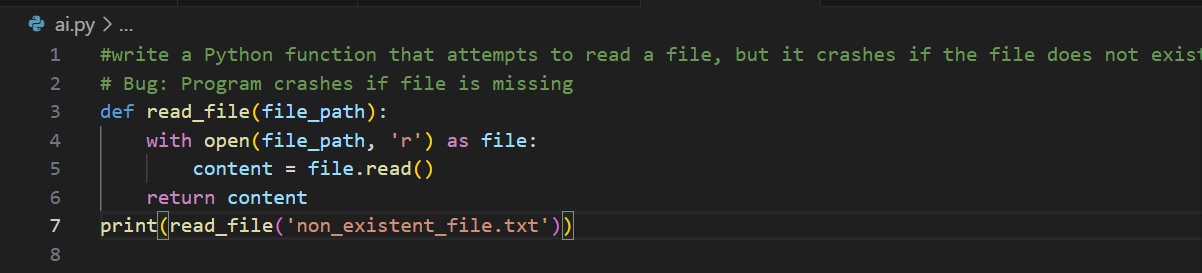
Provide code that attempts to open a non-existent file and crashes.  
Use AI to apply safe error handling.  
# Bug: Program crashes if file is missing  
def read\_file(filename):  
with open(filename, 'r') as f:  
return f.read()  
print(read\_file("nonexistent.txt"))  
Requirements:  
• Implement a try-except block suggested by AI.  
• Add a user-friendly error message.  
• Test with at least 3 scenarios: file exists, file missing, invalid  
path.  
Expected Output #3:  
• Safe file handling with exception management.

PROMPT:

#write a Python function that attempts to read a file, but it crashes if the file does not exist

# Bug: Program crashes if file is missing

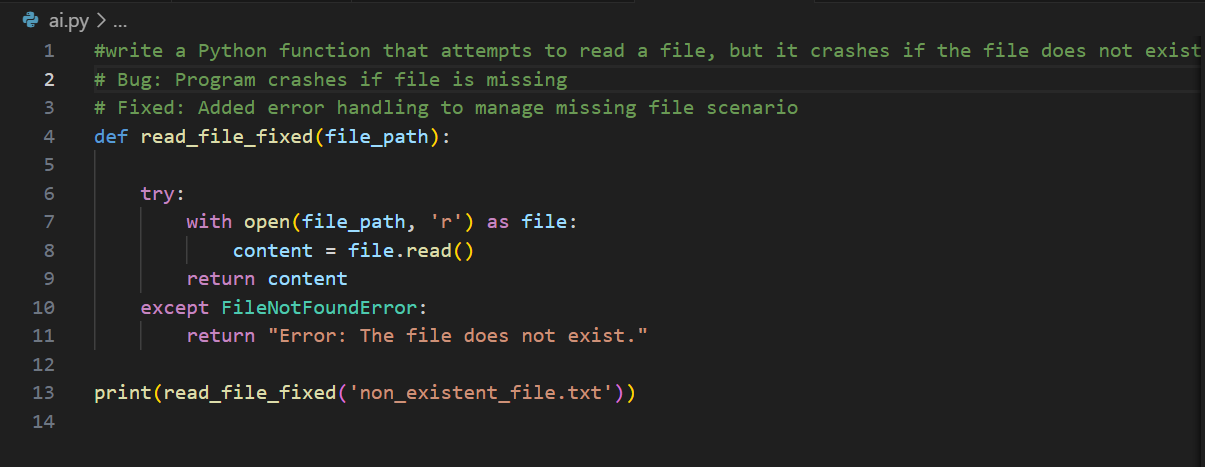
Code:



Error :

FileNotFoundError: [Errno 2] No such file or directory: 'nonexistent.txt'

Correction in code:



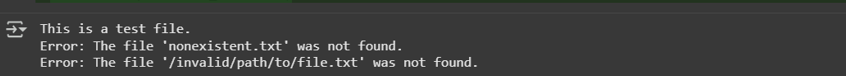
Explanation :

Using try-except to catch FileNotFoundError and other exceptions to prevent the program from crashing.

**Expected behavior:**

* File content prints if file exists.
* User-friendly error message prints if file missing or invalid path.
* No uncaught exceptions.

OUTPUT :



TASK-4:

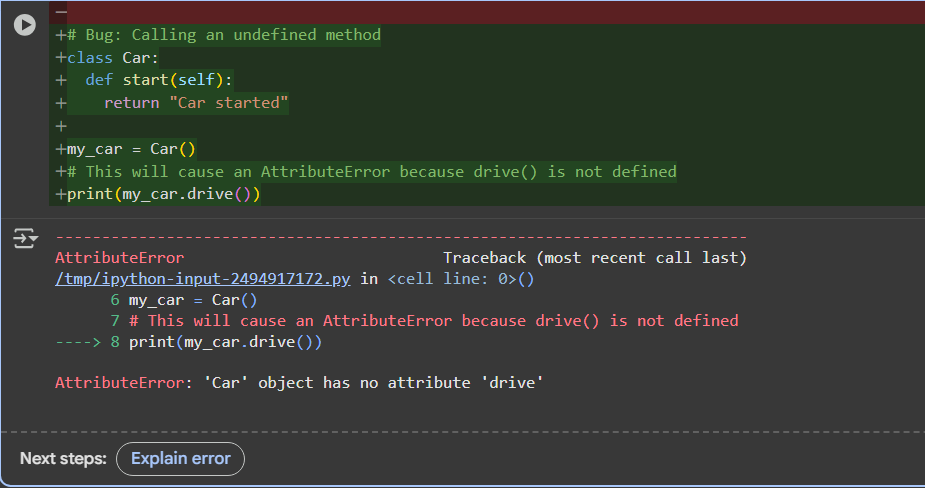
Give a class where a non-existent method is called (e.g.,  
obj.undefined\_method()). Use AI to debug and fix.  
# Bug: Calling an undefined method  
class Car:  
def start(self):  
return "Car started"  
my\_car = Car()  
print(my\_car.drive()) # drive() is not defined  
Requirements:  
• Students must analyze whether to define the missing method or  
correct the method call.  
• Use 3 assert tests to confirm the corrected class works.  
Expected Output #4:  
• Corrected class with clear AI explanation.

PROMPT :

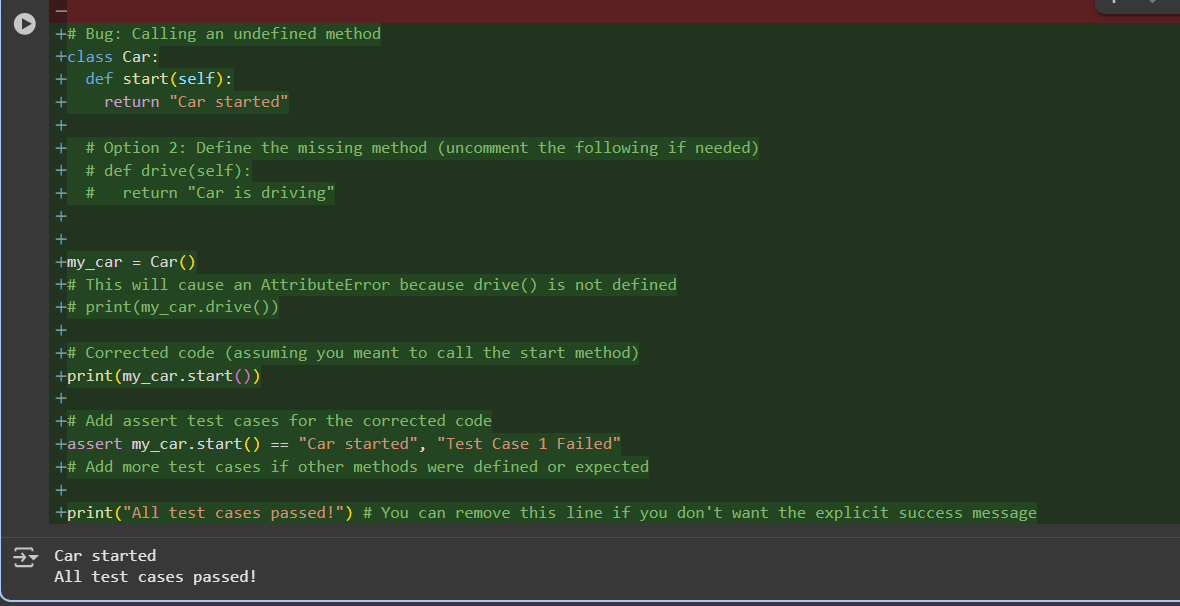
#write a Python class with a bug: a method is being called on an object, but that method **is not defined** in the class.

#Bug: Calling an undefined method

CODE :

****

CORRECTION IN CODE AND OUTPUT :

****

TASK-5 :

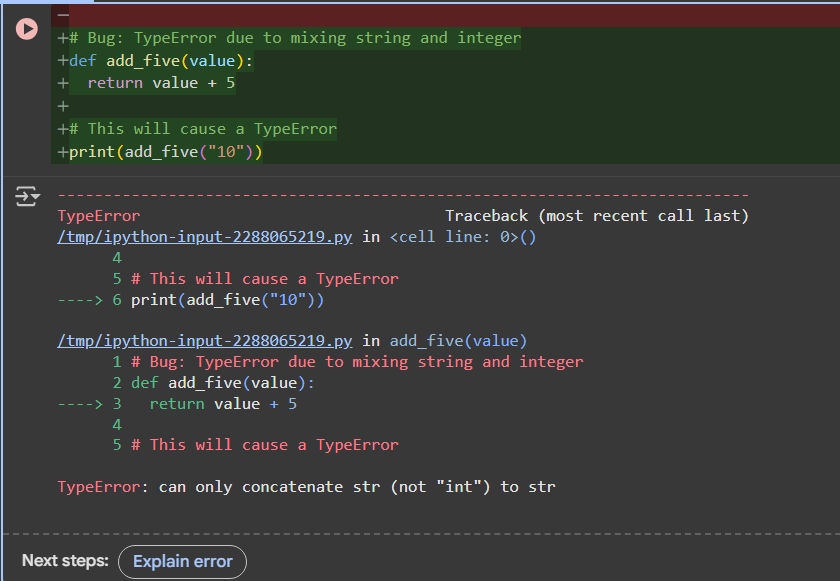
Provide code that adds an integer and string ("5" + 2) causing a  
TypeError. Use AI to resolve the bug.  
# Bug: TypeError due to mixing string and integer  
def add\_five(value):  
return value + 5  
print(add\_five("10"))  
Requirements:  
• Ask AI for two solutions: type casting and string concatenation.  
• Validate with 3 assert test cases.  
Expected Output #5:  
• Corrected code that runs successfully for multiple inputs.  
Note: Report should be submitted a word document for all tasks in a  
single document with prompts, comments & code explanation, and  
output and if required, screenshots  
Evaluation Criteria:  
Criteria Max Marks  
Identification of bugs 0.5  
Application of AI-suggested fixes 0.5  
Explanation and understanding of  
errors 0.5  
Corrected code functionality 0.5  
Report structure and reflection 0.5  
Total 2.5 Marks

PROMPT:

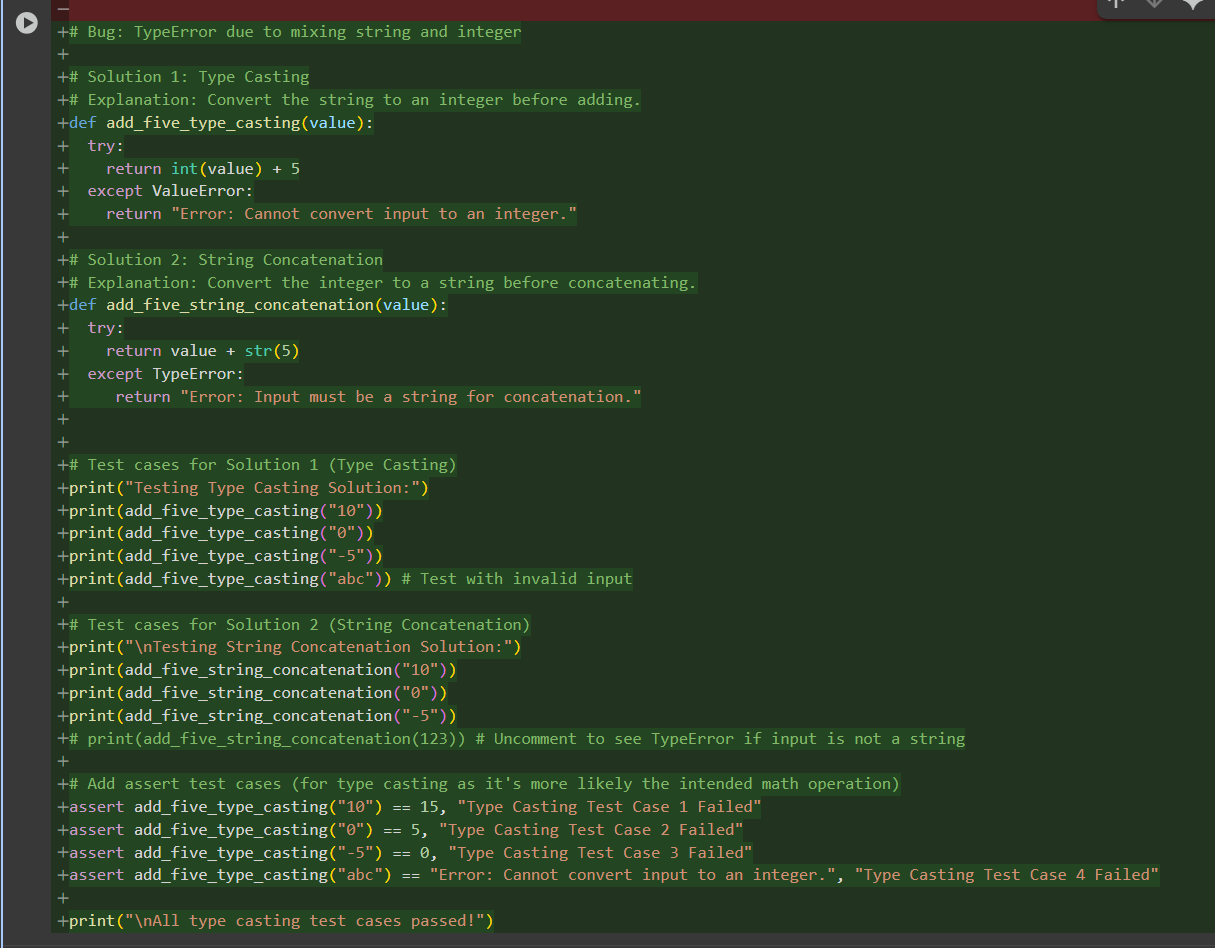
#write a python for the task involving a TypeError from adding a string and an integer.

# Bug: TypeError due to mixing string and integer

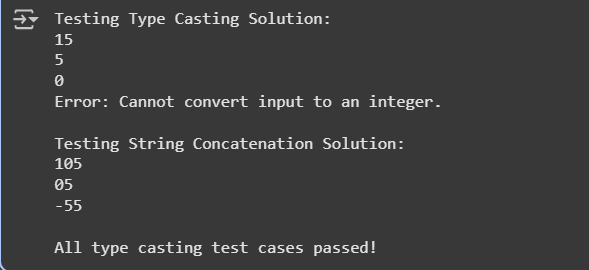
CODE :

****

CORRECTION IN CODE :

****

OUTPUT :

****