Assingment-7.1

Name: Bandana Giri

Hall Ticket No:2303A51590

Batch-07

Lab 7: Error Debugging with AI: Systematic approaches to

finding and fixing bugs

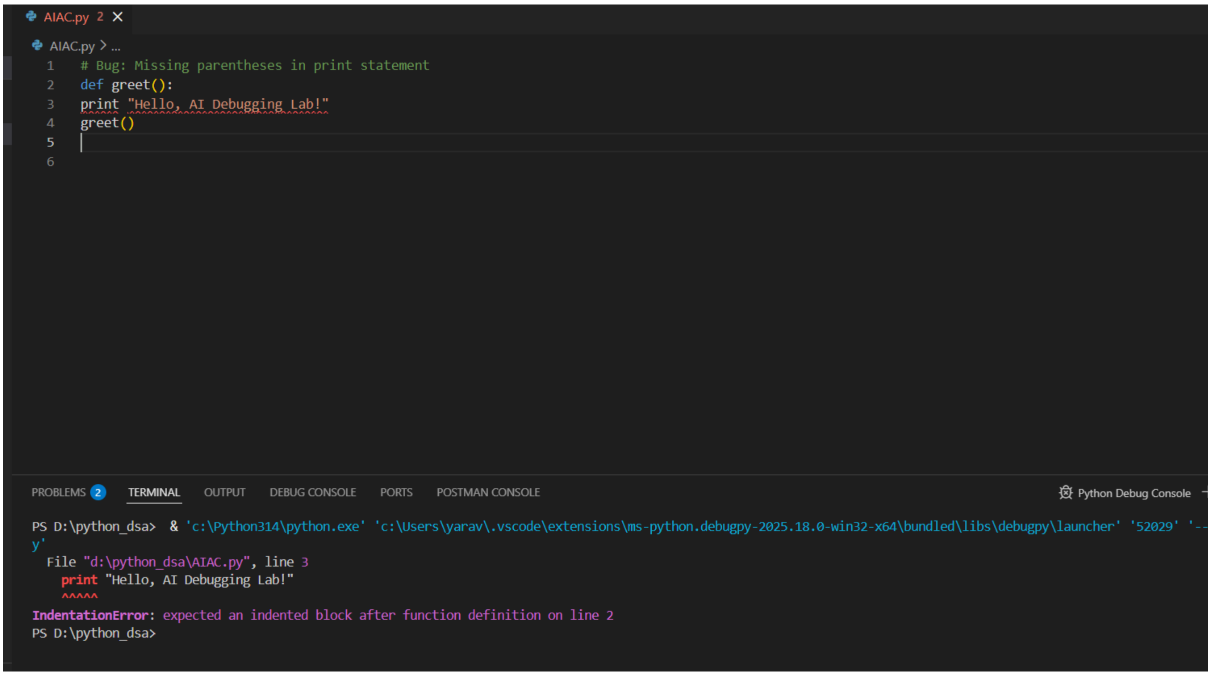
Task Description #1 (Syntax Errors – Missing Parentheses in Print

Statement)

Task: Provide a Python snippet with a missing parenthesis in a print

statement (e.g., print "Hello"). Use AI to detect and fix the syntax

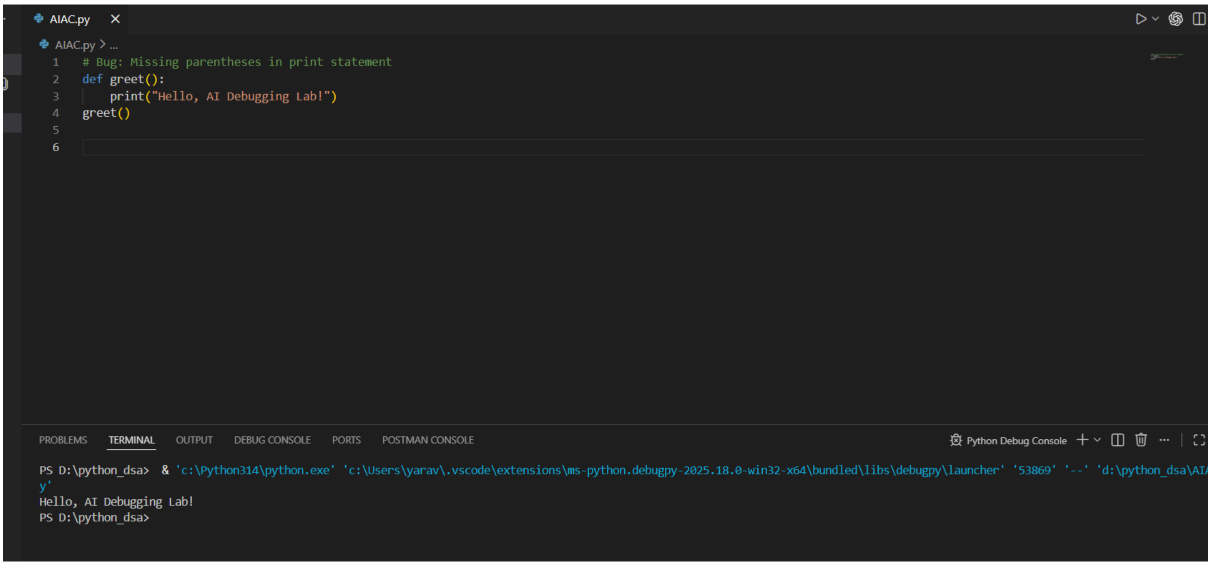
error.



```
1  # Bug: Missing parentheses in print statement
2  def greet():
3  print "Hello, AI Debugging Lab!"
4  greet()
5
6
```

```
PS D:\python_dsa>  & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '52029' '--
y'
  File "d:\python_dsa\AIAC.py", line 3
    print "Hello, AI Debugging Lab!"
    ^^^^^
IndentationError: expected an indented block after function definition on line 2
PS D:\python_dsa>
```

Corrected Code



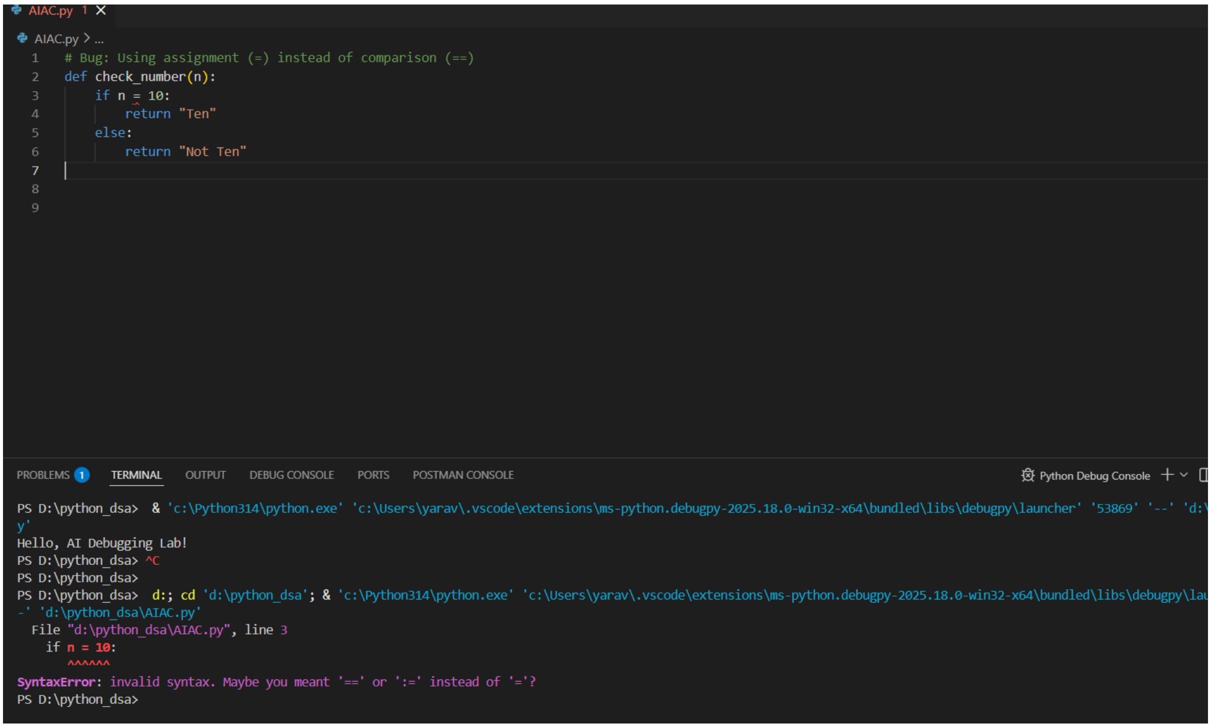
```python
# Bug: Missing parentheses in print statement
def greet():
    print("Hello, AI Debugging Lab!")
greet()
```

```
PS D:\python_dsa>  & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53869' '--' 'd:\python_dsa\AI
y'
Hello, AI Debugging Lab!
PS D:\python_dsa>
```

Task Description #2 (Incorrect condition in an If Statement)

Task: Supply a function where an if-condition mistakenly uses =

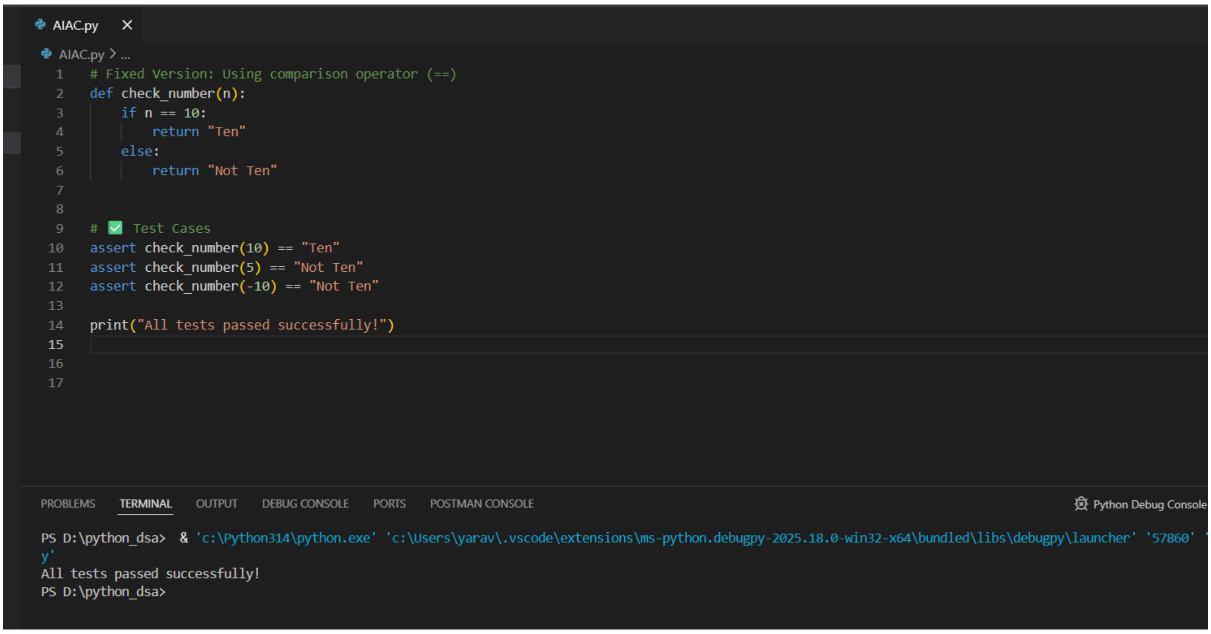
instead of ==. Let AI identify and fix the issue.



```python
# Bug: Using assignment (=) instead of comparison (==)
def check_number(n):
    if n = 10:
        return "Ten"
    else:
        return "Not Ten"
```

```
PS D:\python_dsa>  & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53869' '--' 'd:\
y'
Hello, AI Debugging Lab!
PS D:\python_dsa> ^C
PS D:\python_dsa>
PS D:\python_dsa>  d:; cd 'd:\python_dsa'; & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\lau
-' 'd:\python_dsa\AIAC.py'
  File "d:\python_dsa\AIAC.py", line 3
    if n = 10:
       ^^^^^^
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
PS D:\python_dsa>
```

Corrected Code



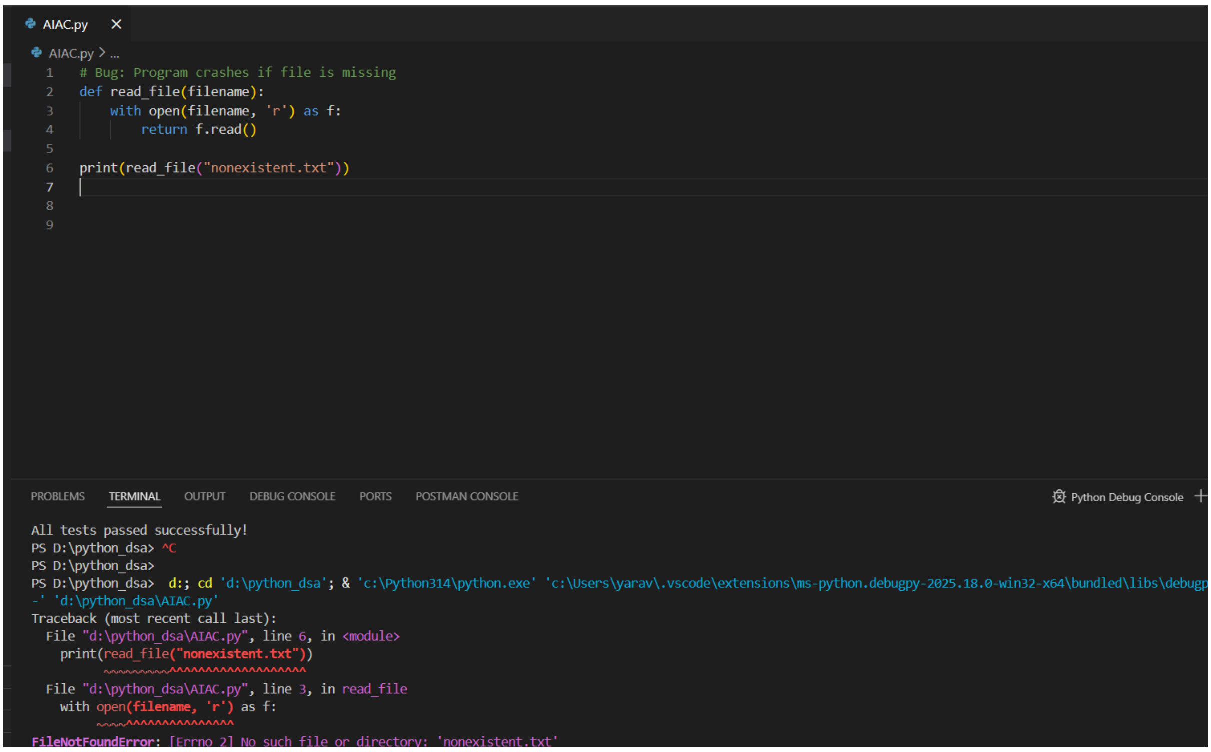
```python
# Fixed Version: Using comparison operator (==)
def check_number(n):
    if n == 10:
        return "Ten"
    else:
        return "Not Ten"


# ✅ Test Cases
assert check_number(10) == "Ten"
assert check_number(5) == "Not Ten"
assert check_number(-10) == "Not Ten"

print("All tests passed successfully!")
```

```
PS D:\python_dsa>  & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '57860' '
y'
All tests passed successfully!
PS D:\python_dsa>
```

Task Description #3 (Runtime Error – File Not Found)

Task: Provide code that attempts to open a non-existent file and

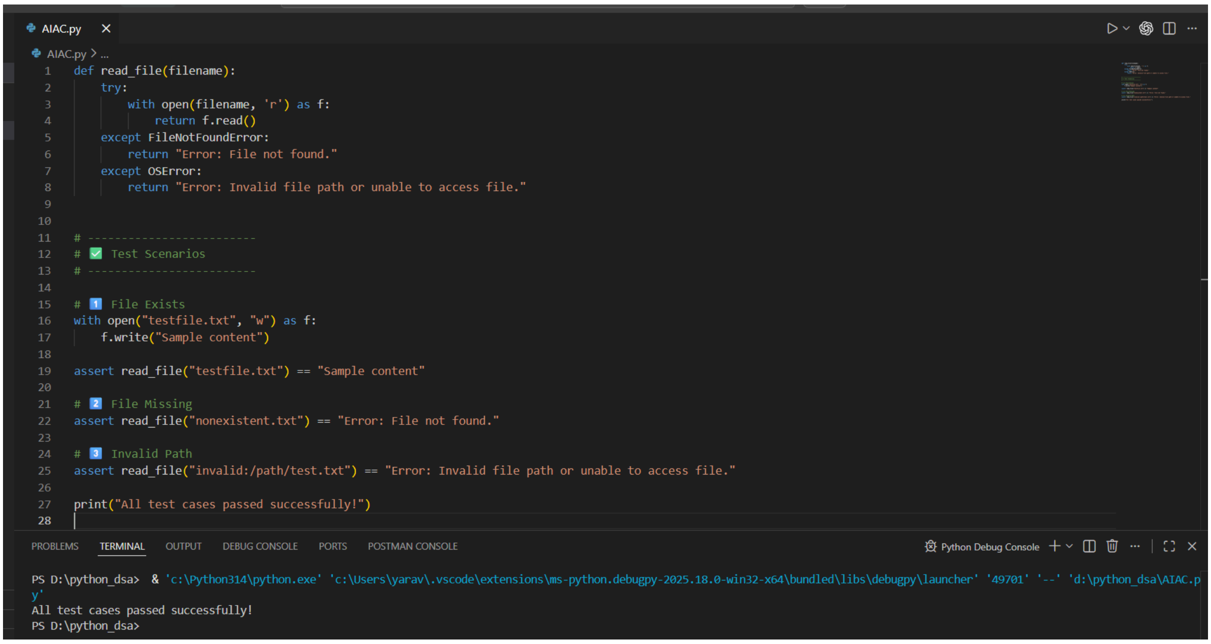
crashes. Use AI to apply safe error handling.



```python
# 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"))
```

```
All tests passed successfully!
PS D:\python_dsa> ^C
PS D:\python_dsa>
PS D:\python_dsa>  d:; cd 'd:\python_dsa'; & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugp
-' 'd:\python_dsa\AIAC.py'
Traceback (most recent call last):
  File "d:\python_dsa\AIAC.py", line 6, in <module>
    print(read_file("nonexistent.txt"))
          ~~~~~~~~~^^^^^^^^^^^^^^^^^^^
  File "d:\python_dsa\AIAC.py", line 3, in read_file
    with open(filename, 'r') as f:
         ~~~~^^^^^^^^^^^^^^^
FileNotFoundError: [Errno 2] No such file or directory: 'nonexistent.txt'
```

Safe Version with Exception Handling

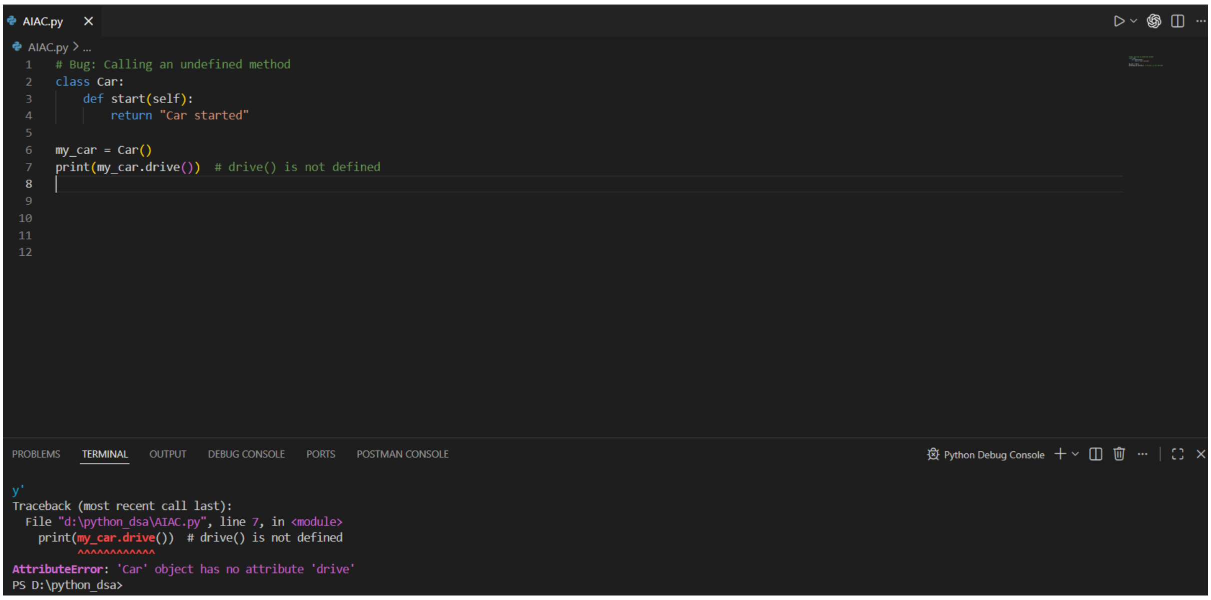


```python
def read_file(filename):
    try:
        with open(filename, 'r') as f:
            return f.read()
    except FileNotFoundError:
        return "Error: File not found."
    except OSError:
        return "Error: Invalid file path or unable to access file."


# ------------------------
# ✅ Test Scenarios
# ------------------------

# 1️⃣ File Exists
with open("testfile.txt", "w") as f:
    f.write("Sample content")

assert read_file("testfile.txt") == "Sample content"

# 2️⃣ File Missing
assert read_file("nonexistent.txt") == "Error: File not found."

# 3️⃣ Invalid Path
assert read_file("invalid:/path/test.txt") == "Error: Invalid file path or unable to access file."

print("All test cases passed successfully!")
```

```
PS D:\python_dsa>  & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '49701' '--' 'd:\python_dsa\AIAC.p
y'
All test cases passed successfully!
PS D:\python_dsa>
```

Task Description #4 (Calling a Non-Existent Method)

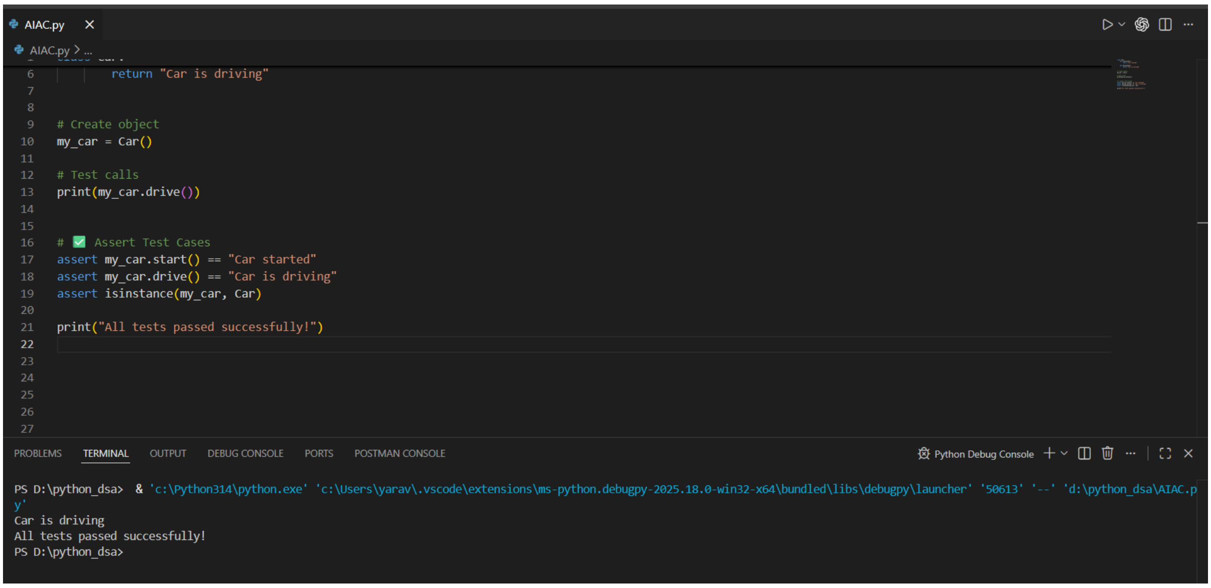
Task: Give a class where a non-existent method is called (e.g., obj.undefined_method()). Use AI to debug and fix.



```python
# Bug: Calling an undefined method
class Car:
    def start(self):
        return "Car started"

my_car = Car()
print(my_car.drive())  # drive() is not defined
```

```
y'
Traceback (most recent call last):
  File "d:\python_dsa\AIAC.py", line 7, in <module>
    print(my_car.drive())  # drive() is not defined
          ^^^^^^^^^^^^
AttributeError: 'Car' object has no attribute 'drive'
PS D:\python_dsa>
```

Corrected Code



```python
        return "Car is driving"


# Create object
my_car = Car()

# Test calls
print(my_car.drive())


# ✅ Assert Test Cases
assert my_car.start() == "Car started"
assert my_car.drive() == "Car is driving"
assert isinstance(my_car, Car)

print("All tests passed successfully!")
```

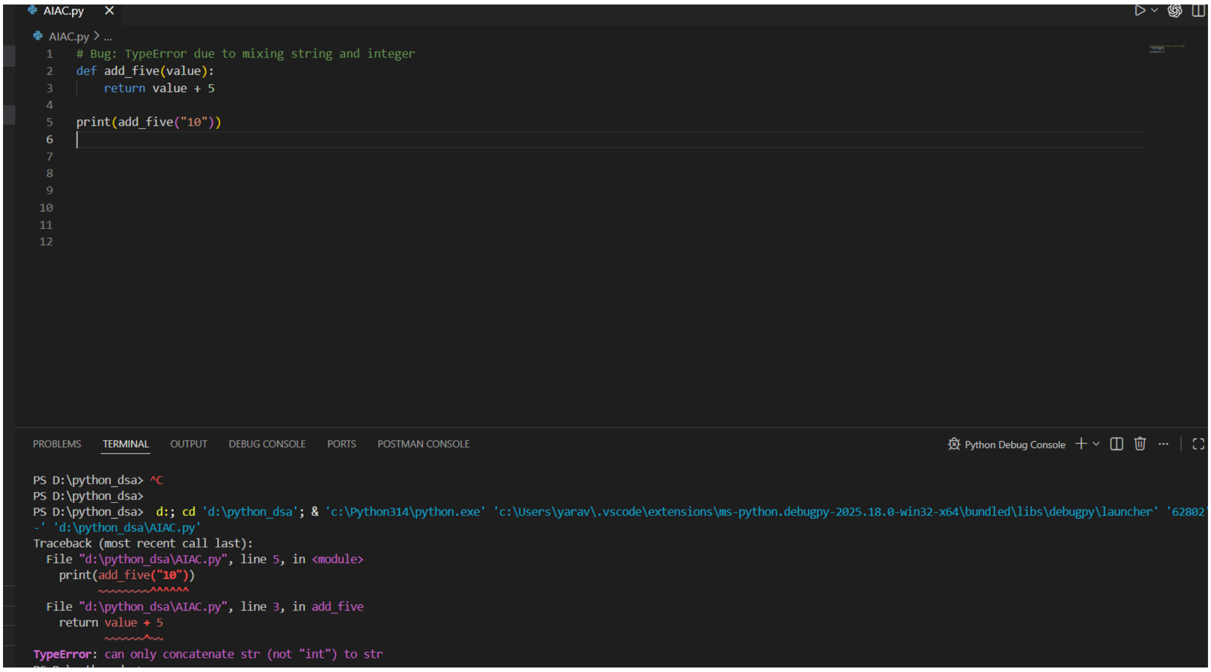
```
PS D:\python_dsa>  & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '50613' '--' 'd:\python_dsa\AIAC.p
y'
Car is driving
All tests passed successfully!
PS D:\python_dsa>
```

Task Description #5 (TypeError – Mixing Strings and Integers in

Addition)

Task: Provide code that adds an integer and string ("5" + 2) causing

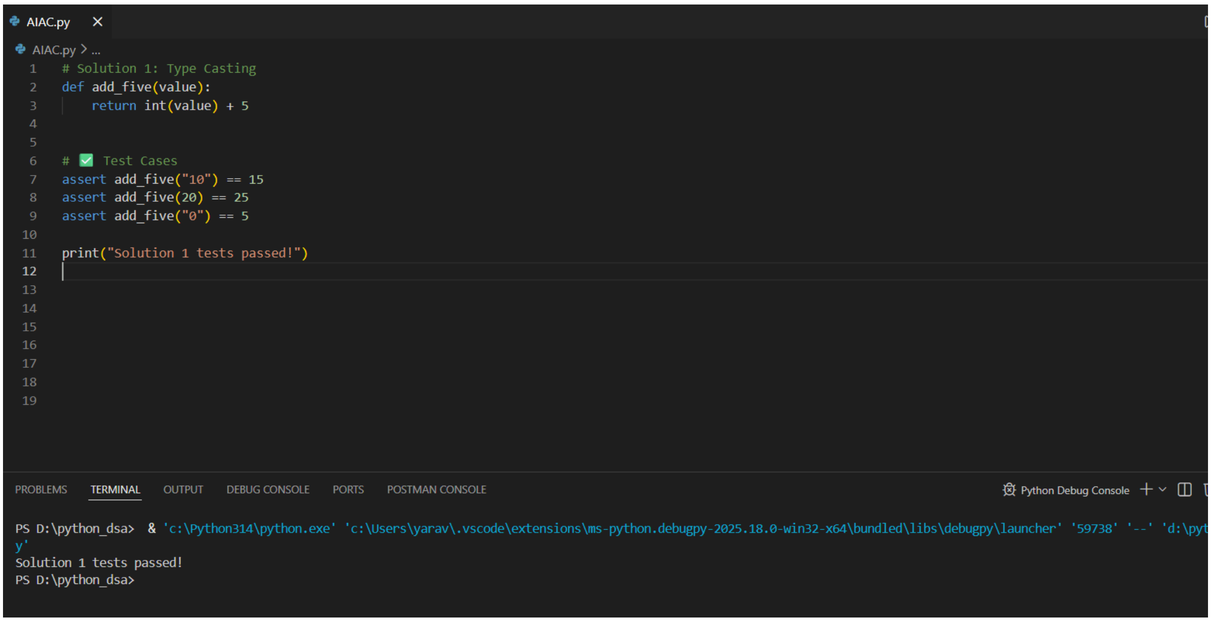
a TypeError. Use AI to resolve the bug



```python
# Bug: TypeError due to mixing string and integer
def add_five(value):
    return value + 5

print(add_five("10"))
```

```
PS D:\python_dsa> ^C
PS D:\python_dsa>
PS D:\python_dsa>  d:; cd 'd:\python_dsa'; & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '62802'
-' 'd:\python_dsa\AIAC.py'
Traceback (most recent call last):
  File "d:\python_dsa\AIAC.py", line 5, in <module>
    print(add_five("10"))
          ~~~~~~~~^^^^^^
  File "d:\python_dsa\AIAC.py", line 3, in add_five
    return value + 5
           ~~~~~~^~~
TypeError: can only concatenate str (not "int") to str
```

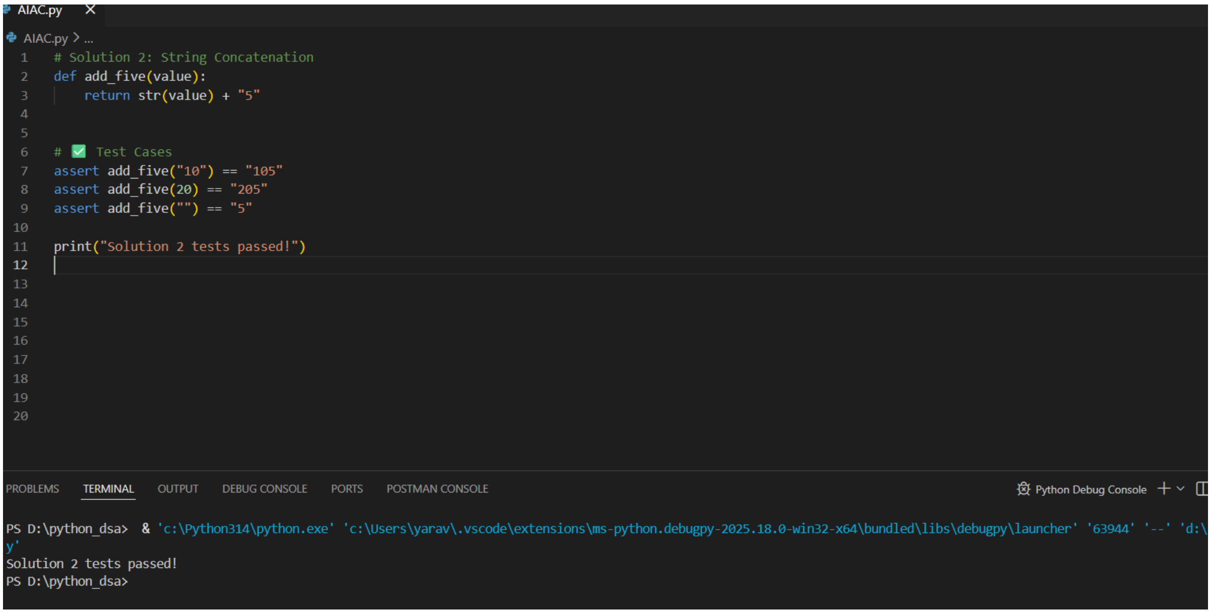
Solution 1: Type Casting (Convert to Integer)



```python
# Solution 1: Type Casting
def add_five(value):
    return int(value) + 5


# ✅ Test Cases
assert add_five("10") == 15
assert add_five(20) == 25
assert add_five("0") == 5

print("Solution 1 tests passed!")
```

```
PS D:\python_dsa>  & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '59738' '--' 'd:\pyt
y'
Solution 1 tests passed!
PS D:\python_dsa>
```

Solution 2: String Concatenation



```python
# Solution 2: String Concatenation
def add_five(value):
    return str(value) + "5"


# ✅ Test Cases
assert add_five("10") == "105"
assert add_five(20) == "205"
assert add_five("") == "5"

print("Solution 2 tests passed!")
```

```
PS D:\python_dsa>  & 'c:\Python314\python.exe' 'c:\Users\yarav\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '63944' '--' 'd:\
y'
Solution 2 tests passed!
PS D:\python_dsa>
```