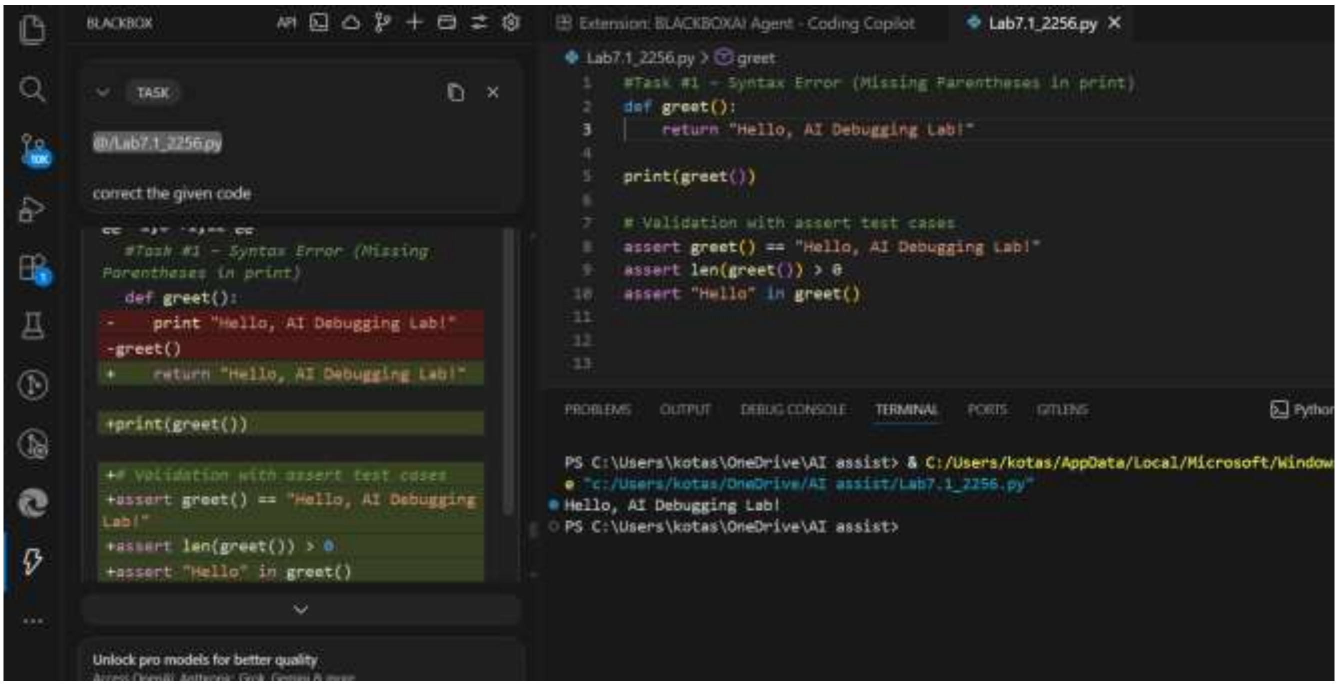
Lab Assignment- 7.1

NAME: A.Harshita

ID.NO:2303A52211

Subject: AI Assisted Coding

Task #1 – Syntax Error (Missing Parentheses in print)
Prompt: Use AI to detect the syntax error caused by missing parentheses in the print statement, correct it, and validate the fix using at least 3 assert test cases.



correct the given code

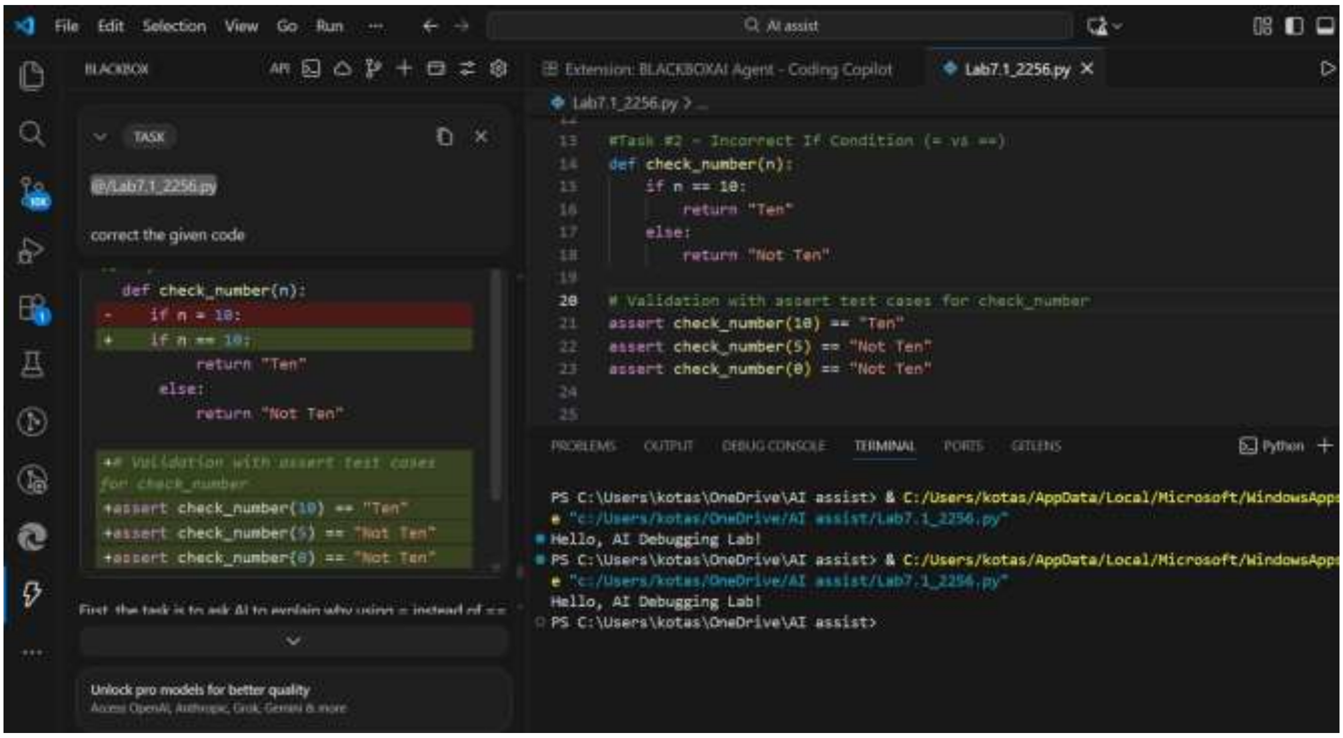
```diff
   #Task #1 - Syntax Error (Missing
Parentheses in print)
   def greet():
-      print "Hello, AI Debugging Lab!"
-greet()
+      return "Hello, AI Debugging Lab!"

+print(greet())

+# Validation with assert test cases
+assert greet() == "Hello, AI Debugging
Lab!"
+assert len(greet()) > 0
+assert "Hello" in greet()
```

```python
#Task #1 - Syntax Error (Missing Parentheses in print)
def greet():
    return "Hello, AI Debugging Lab!"

print(greet())

# Validation with assert test cases
assert greet() == "Hello, AI Debugging Lab!"
assert len(greet()) > 0
assert "Hello" in greet()
```

```
PS C:\Users\kotas\OneDrive\AI assist> & C:/Users/kotas/AppData/Local/Microsoft/Window
"c:/Users/kotas/OneDrive/AI assist/Lab7.1_2256.py"
Hello, AI Debugging Lab!
PS C:\Users\kotas\OneDrive\AI assist>
```

AI Explanation
Python 3 requires parentheses in print(). Without them, it raises a **SyntaxError**.
Code Explanation
The syntax was corrected by adding parentheses to the print function.

Task #2 – Incorrect If Condition (= vs ==)
Prompt
Ask AI to explain why using = instead of == causes a bug, correct the code, and verify the fix using 3 assert test cases.


File Edit Selection View Go Run
AI assist
BLACKBOX
TASK
Extension: BLACKBOXAI Agent - Coding Copilot
Lab7.1_2256.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
Python
Unlock pro models for better quality
Access OpenAI, Anthropic, Grok, Gemini & more

@/Lab7.1_2256.py

correct the given code

```
 def check_number(n):
-    if n = 10:
+    if n == 10:
         return "Ten"
     else:
         return "Not Ten"

+# Validation with assert test cases for check_number
+assert check_number(10) == "Ten"
+assert check_number(5) == "Not Ten"
+assert check_number(0) == "Not Ten"
```

First, the task is to ask AI to explain why using = instead of ==

```
#Task #2 - Incorrect If Condition (= vs ==)
def check_number(n):
    if n == 10:
        return "Ten"
    else:
        return "Not Ten"

# Validation with assert test cases for check_number
assert check_number(10) == "Ten"
assert check_number(5) == "Not Ten"
assert check_number(0) == "Not Ten"
```

```
PS C:\Users\kotas\OneDrive\AI assist> & C:/Users/kotas/AppData/Local/Microsoft/WindowsApps
 "c:/Users/kotas/OneDrive/AI assist/Lab7.1_2256.py"
Hello, AI Debugging Lab!
PS C:\Users\kotas\OneDrive\AI assist> & C:/Users/kotas/AppData/Local/Microsoft/WindowsApps
 "c:/Users/kotas/OneDrive/AI assist/Lab7.1_2256.py"
Hello, AI Debugging Lab!
PS C:\Users\kotas\OneDrive\AI assist>
```

AI Explanation

= is an assignment operator, not comparison. Conditions require ==.

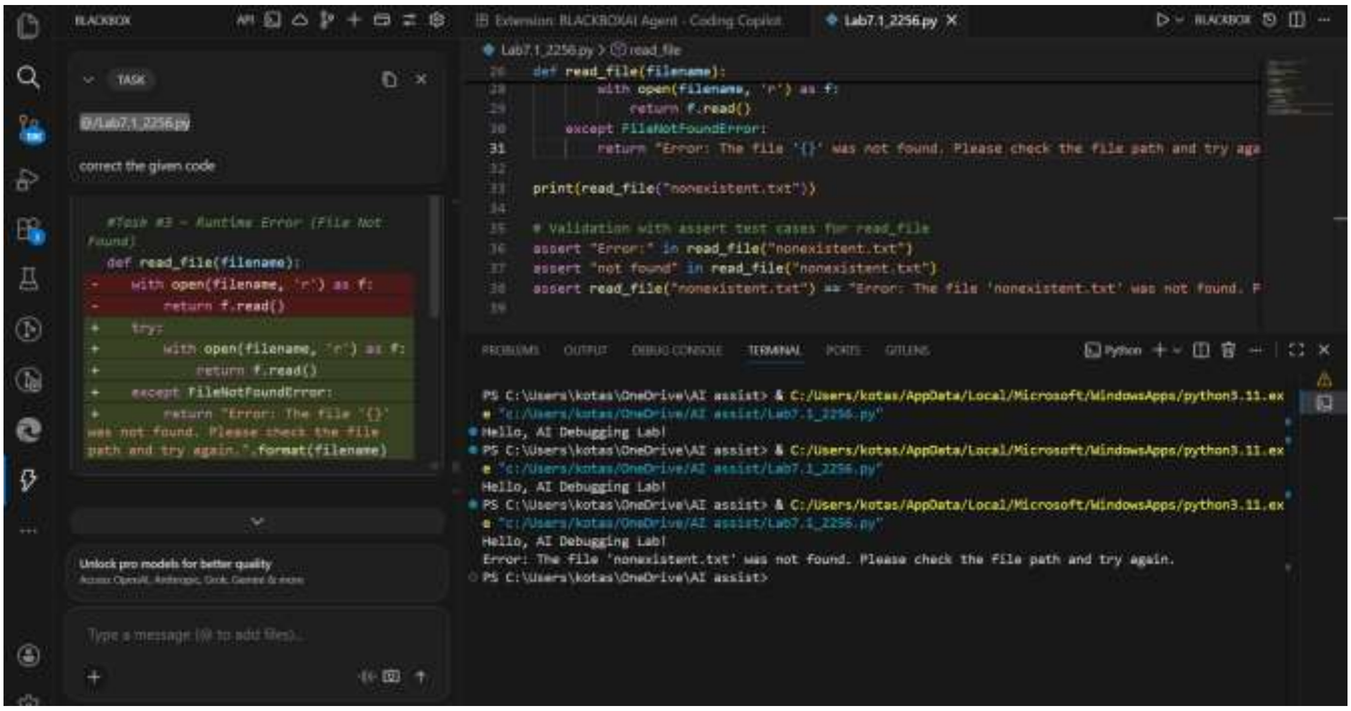
Code Explanation

Using == ensures proper comparison instead of assignment.

Task #3 – Runtime Error (File Not Found)

Prompt

Use AI to add safe error handling using try-except, provide a user-friendly message, and validate with 3 test scenarios and assert checks.


BLACKBOX
TASK
Extension: BLACKBOXAI Agent - Coding Copilot
Lab7.1_2256.py
Lab7.1_2256.py > read_file
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
Python
Unlock pro models for better quality
Access OpenAI, Anthropic, Grok, Gemini & more
Type a message (@ to add files)...

@/Lab7.1_2256.py

correct the given code

```
 #Task #3 - Runtime Error (File Not Found)
 def read_file(filename):
-    with open(filename, 'r') as f:
-        return f.read()
+    try:
+        with open(filename, 'r') as f:
+            return f.read()
+    except FileNotFoundError:
+        return "Error: The file '{}' was not found. Please check the file path and try again.".format(filename)
```

```
def read_file(filename):
        with open(filename, 'r') as f:
            return f.read()
    except FileNotFoundError:
        return "Error: The file '{}' was not found. Please check the file path and try aga

print(read_file("nonexistent.txt"))

# Validation with assert test cases for read_file
assert "Error:" in read_file("nonexistent.txt")
assert "not found" in read_file("nonexistent.txt")
assert read_file("nonexistent.txt") == "Error: The file 'nonexistent.txt' was not found. P
```

```
PS C:\Users\kotas\OneDrive\AI assist> & C:/Users/kotas/AppData/Local/Microsoft/WindowsApps/python3.11.ex
 "c:/Users/kotas/OneDrive/AI assist/Lab7.1_2256.py"
Hello, AI Debugging Lab!
PS C:\Users\kotas\OneDrive\AI assist> & C:/Users/kotas/AppData/Local/Microsoft/WindowsApps/python3.11.ex
 "c:/Users/kotas/OneDrive/AI assist/Lab7.1_2256.py"
Hello, AI Debugging Lab!
PS C:\Users\kotas\OneDrive\AI assist> & C:/Users/kotas/AppData/Local/Microsoft/WindowsApps/python3.11.ex
 "c:/Users/kotas/OneDrive/AI assist/Lab7.1_2256.py"
Hello, AI Debugging Lab!
Error: The file 'nonexistent.txt' was not found. Please check the file path and try again.
PS C:\Users\kotas\OneDrive\AI assist>
```

AI Explanation

The error occurs because the program tries to open a non-existent file, so AI suggests using a try–except block to handle the exception safely.

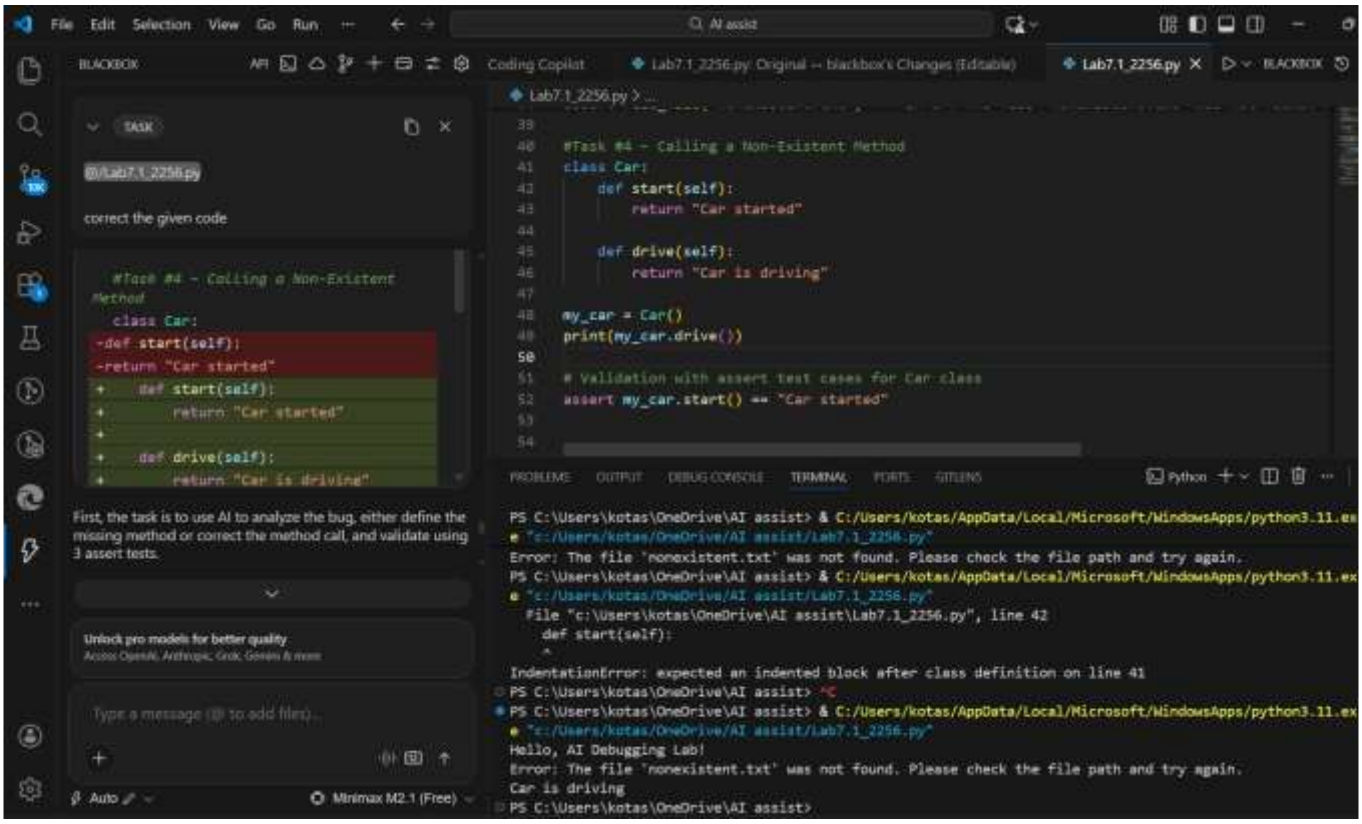
Code Explanation

try-except prevents crashes and handles runtime errors gracefully.

Task #4 – Calling a Non-Existent Method

Prompt

Use AI to analyze the bug, either define the missing method or correct the method call, and validate using 3 assert tests.



@/Lab7.1_2256.py

correct the given code

```diff
  #Task #4 - Calling a Non-Existent Method
  class Car:
-def start(self):
-return "Car started"
+    def start(self):
+        return "Car started"
+
+    def drive(self):
+        return "Car is driving"
```

First, the task is to use AI to analyze the bug, either define the missing method or correct the method call, and validate using 3 assert tests.

```python
#Task #4 - Calling a Non-Existent Method
class Car:
    def start(self):
        return "Car started"

    def drive(self):
        return "Car is driving"

my_car = Car()
print(my_car.drive())

# Validation with assert test cases for Car class
assert my_car.start() == "Car started"
```

```
PS C:\Users\kotas\OneDrive\AI assist> & C:/Users/kotas/AppData/Local/Microsoft/WindowsApps/python3.11.ex
"c:/Users/kotas/OneDrive/AI assist/Lab7.1_2256.py"
Error: The file 'nonexistent.txt' was not found. Please check the file path and try again.
PS C:\Users\kotas\OneDrive\AI assist> & C:/Users/kotas/AppData/Local/Microsoft/WindowsApps/python3.11.ex
"c:/Users/kotas/OneDrive/AI assist/Lab7.1_2256.py"
  File "c:\Users\kotas\OneDrive\AI assist\Lab7.1_2256.py", line 42
    def start(self):
    ^
IndentationError: expected an indented block after class definition on line 41
PS C:\Users\kotas\OneDrive\AI assist> ^C
PS C:\Users\kotas\OneDrive\AI assist> & C:/Users/kotas/AppData/Local/Microsoft/WindowsApps/python3.11.ex
"c:/Users/kotas/OneDrive/AI assist/Lab7.1_2256.py"
Hello, AI Debugging Lab!
Error: The file 'nonexistent.txt' was not found. Please check the file path and try again.
Car is driving
PS C:\Users\kotas\OneDrive\AI assist>
```

File
Edit
Selection
View
Go
Run
AI assist
BLACKBOX
TASK
Coding Copilot
Lab7.1_2256.py: Original ↔ blackbox's Changes (Editable)
Lab7.1_2256.py
Unlock pro models for better quality
Access OpenAI, Anthropic, Grok, Gemini & more
Type a message (@ to add files)...
Auto
Minimax M2.1 (Free)
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
GITLENS
Python

AI Explanation

drive() does not exist, causing **AttributeError**.

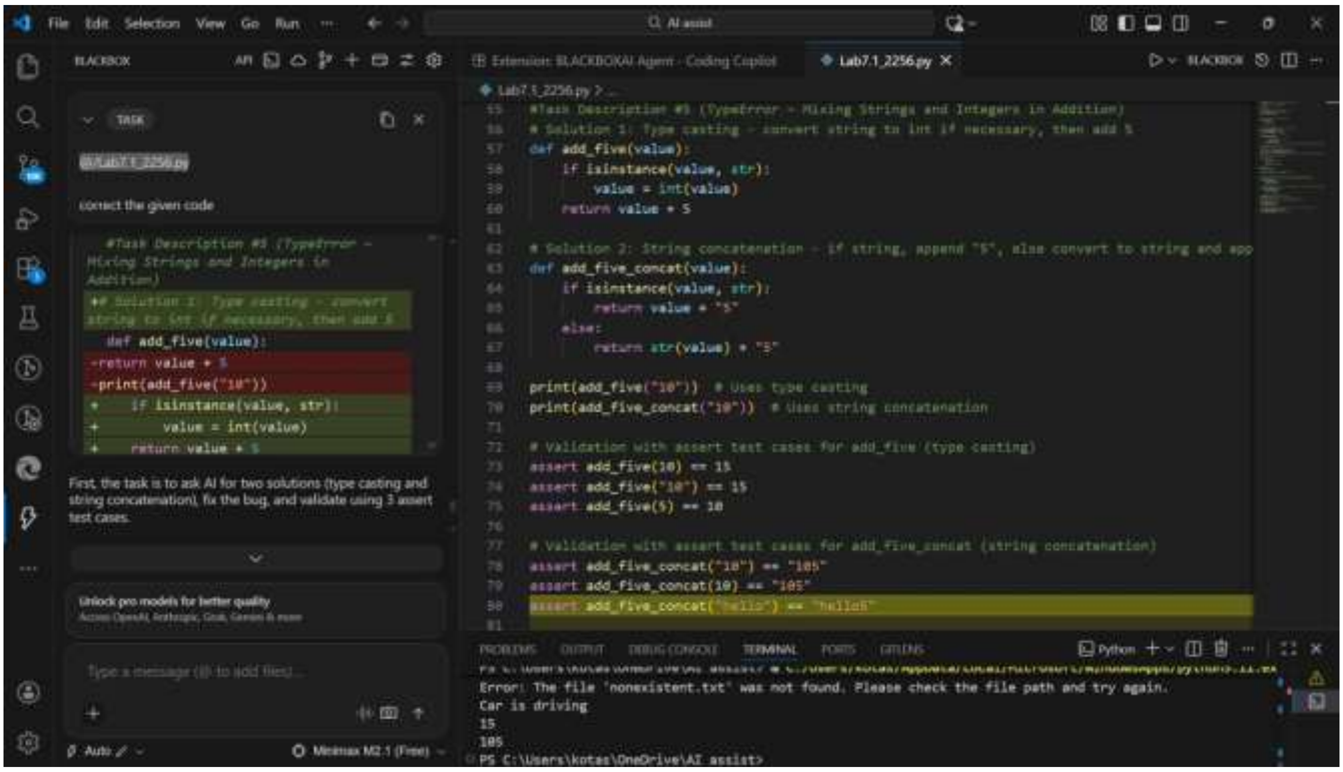
Code Explanation

The missing method was defined to eliminate the AttributeError.

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

Prompt

Ask AI for two solutions (type casting and string concatenation), fix the bug, and validate using 3 assert test cases.



File Edit Selection View Go Run
AI assist
BLACKBOX
Extension: BLACKBOXAI Agent - Coding Copilot
Lab7.1_2256.py
TASK
@/Lab7.1_2256.py

correct the given code

```
#Task Description #5 (TypeError - Mixing Strings and Integers in Addition)
+# Solution 1: Type casting - convert string to int if necessary, then add 5
 def add_five(value):
-return value + 5
-print(add_five("10"))
+    if isinstance(value, str):
+        value = int(value)
+    return value + 5
```

First, the task is to ask AI for two solutions (type casting and string concatenation), fix the bug, and validate using 3 assert test cases.

Unlock pro models for better quality
Access OpenAI, Anthropic, Grok, Gemini & more
Type a message (@ to add files)...
Auto
Minimax M2.1 (Free)

```python
#Task Description #5 (TypeError - Mixing Strings and Integers in Addition)
# Solution 1: Type casting - convert string to int if necessary, then add 5
def add_five(value):
    if isinstance(value, str):
        value = int(value)
    return value + 5

# Solution 2: String concatenation - if string, append "5", else convert to string and app
def add_five_concat(value):
    if isinstance(value, str):
        return value + "5"
    else:
        return str(value) + "5"

print(add_five("10"))  # Uses type casting
print(add_five_concat("10"))  # Uses string concatenation

# Validation with assert test cases for add_five (type casting)
assert add_five(10) == 15
assert add_five("10") == 15
assert add_five(5) == 10

# Validation with assert test cases for add_five_concat (string concatenation)
assert add_five_concat("10") == "105"
assert add_five_concat(10) == "105"
assert add_five_concat("hello") == "hello5"
```

```
Error: The file 'nonexistent.txt' was not found. Please check the file path and try again.
Car is driving
15
105
PS C:\Users\kotas\OneDrive\AI assist>
```

AI Explanation

The error happens due to adding incompatible data types, so AI suggests type casting or string concatenation to fix it.

Code Explanation

The error occurred due to mixing data types. Fixes include converting to integer or concatenating strings.