2303A51797

BATCH-26

ASSIGNMENT-7.3

Task 1: Fixing Syntax Errors

Scenario

You are reviewing a Python program where a basic function definition contains a syntax error.

Requirements

• Provide a Python function add(a, b) with a missing colon

• Use an AI tool to detect the syntax error

• Allow AI to correct the function definition

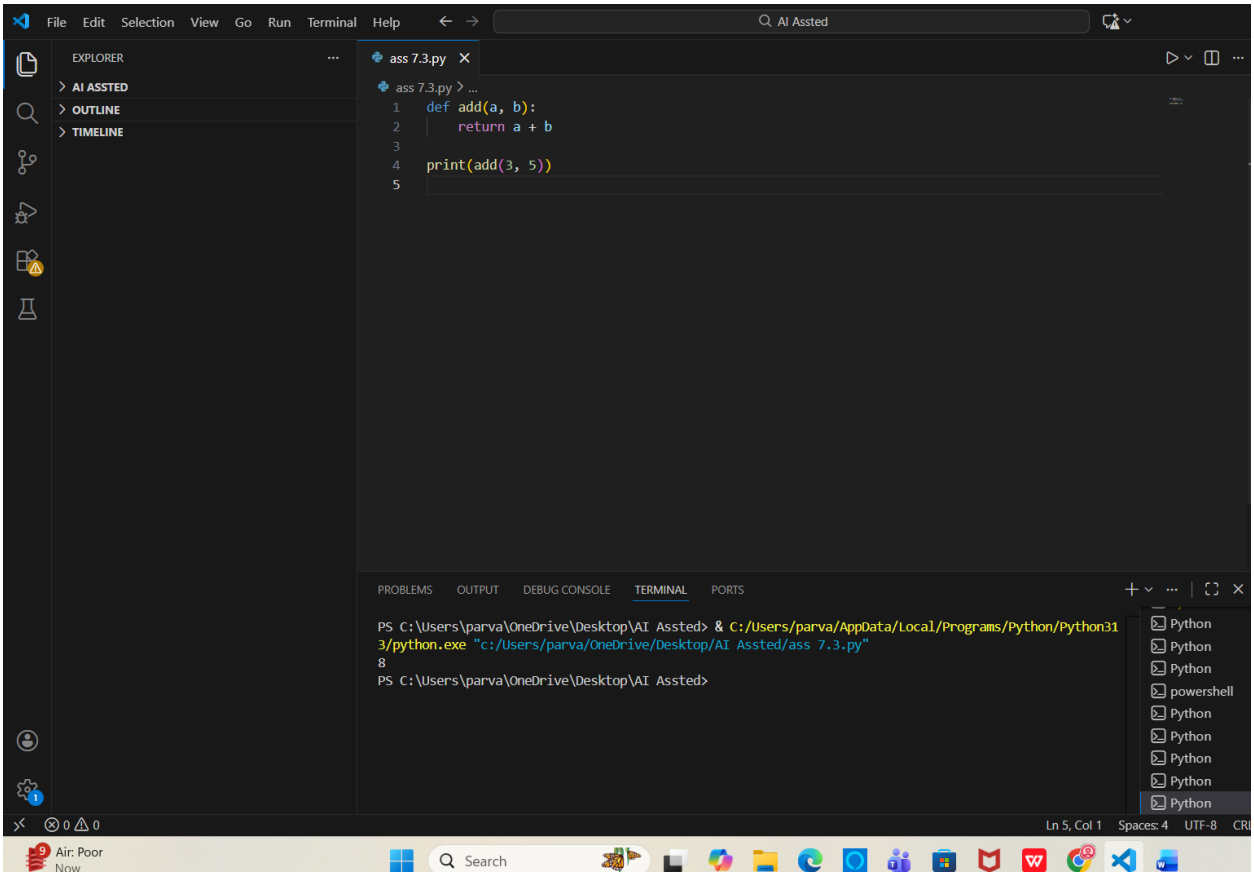
• Observe how AI explains the syntax issue

Expected Output

• Corrected function with proper syntax

• Syntax error resolved successfully

• AI-generated explanation of the fix



Task 2: Debugging Logic Errors in Loops

Scenario

You are debugging a loop that runs infinitely due to a logical mistake.

Requirements

• Provide a loop with an increment or decrement error

• Use AI to identify the cause of infinite iteration

• Let AI fix the loop logic

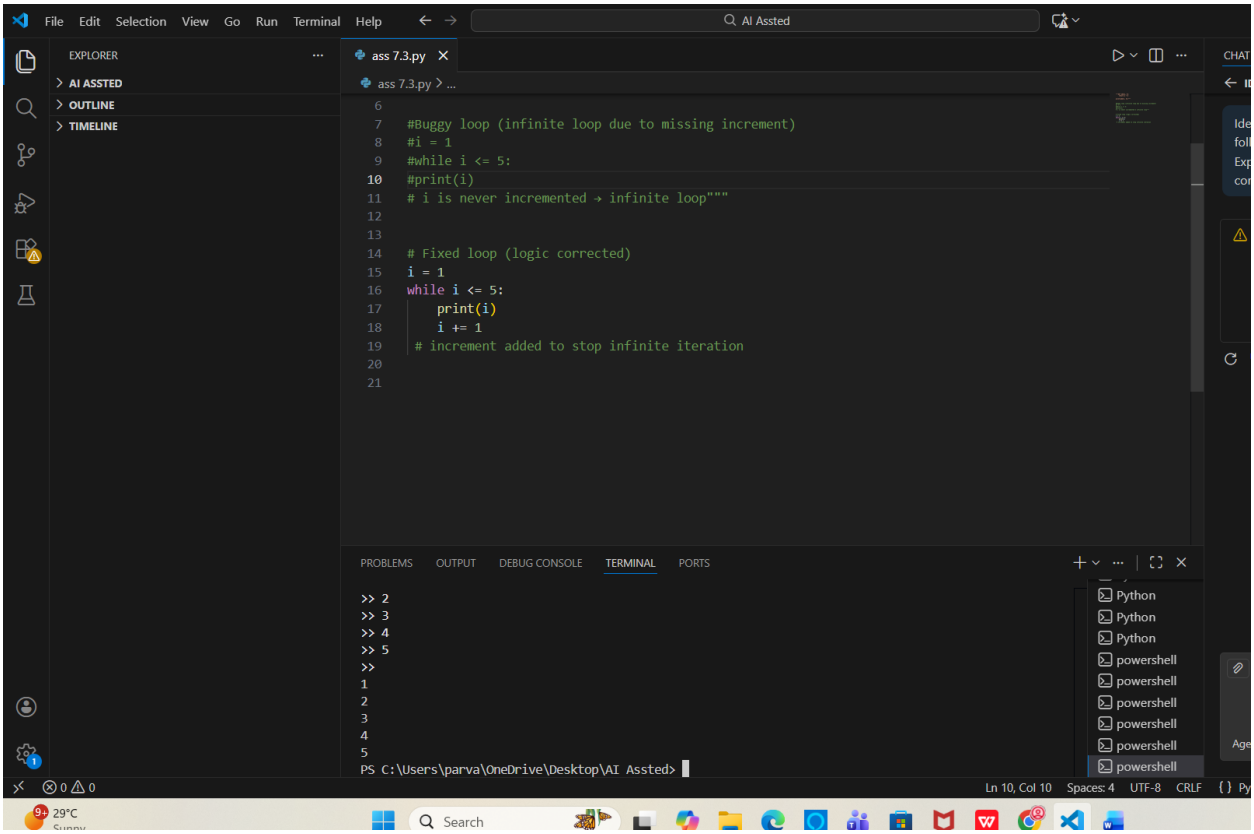
• Analyze the corrected loop behavior

Expected Output

• Infinite loop issue resolved

• Correct increment/decrement logic applied

• AI explanation of the logic error



Task 3: Handling Runtime Errors (Division by Zero)

Scenario

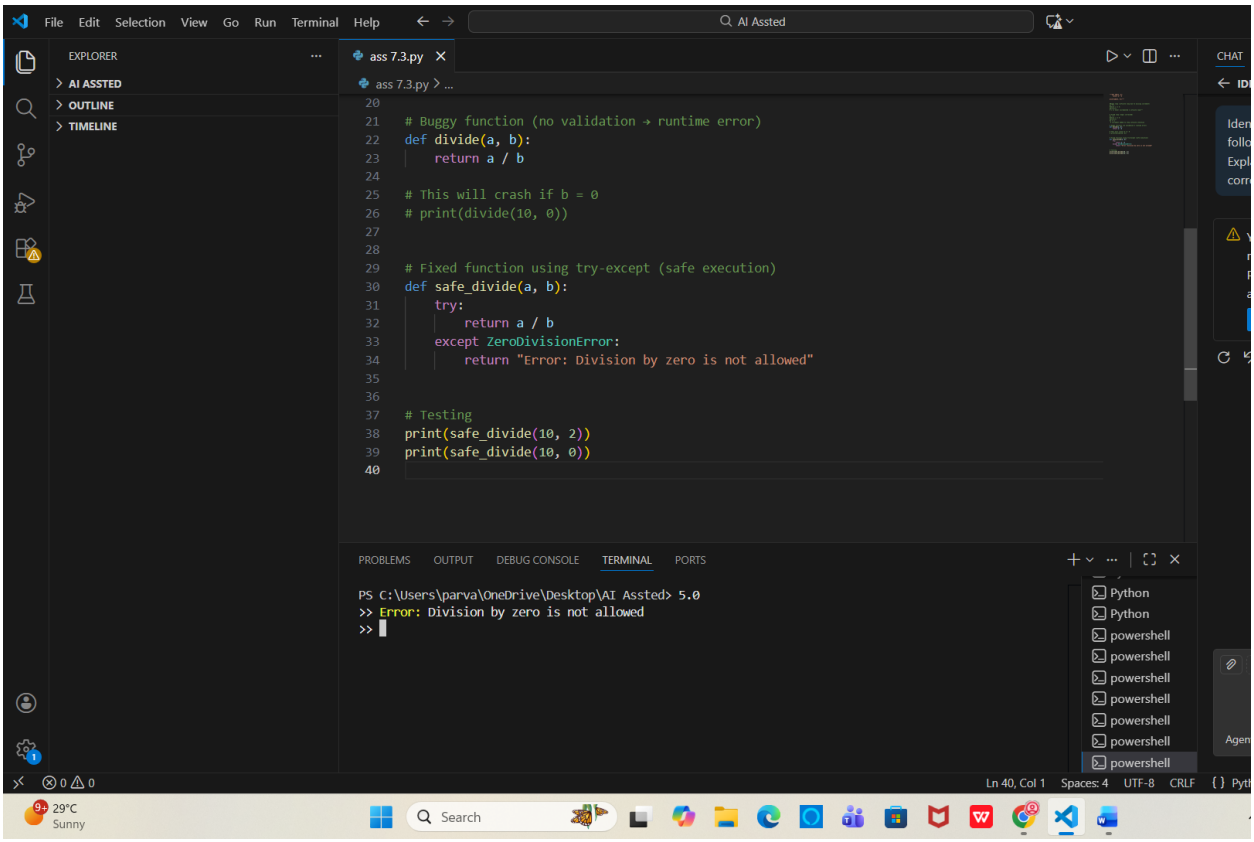
A Python function crashes during execution due to a division by zero error.

Requirements

- Provide a function that performs division without validation
- Use AI to identify the runtime error
- Let AI add try-except blocks for safe execution
- Review AI's error-handling approach

Expected Output

- Function executes safely without crashing
- Division by zero handled using try-except
- Clear AI-generated explanation of runtime error handling



Task 4: Debugging Class Definition Errors

Scenario

You are given a faulty Python class where the constructor is incorrectly defined.

Requirements

• Provide a class definition with missing self-parameter

• Use AI to identify the issue in the __init__() method

• Allow AI to correct the class definition

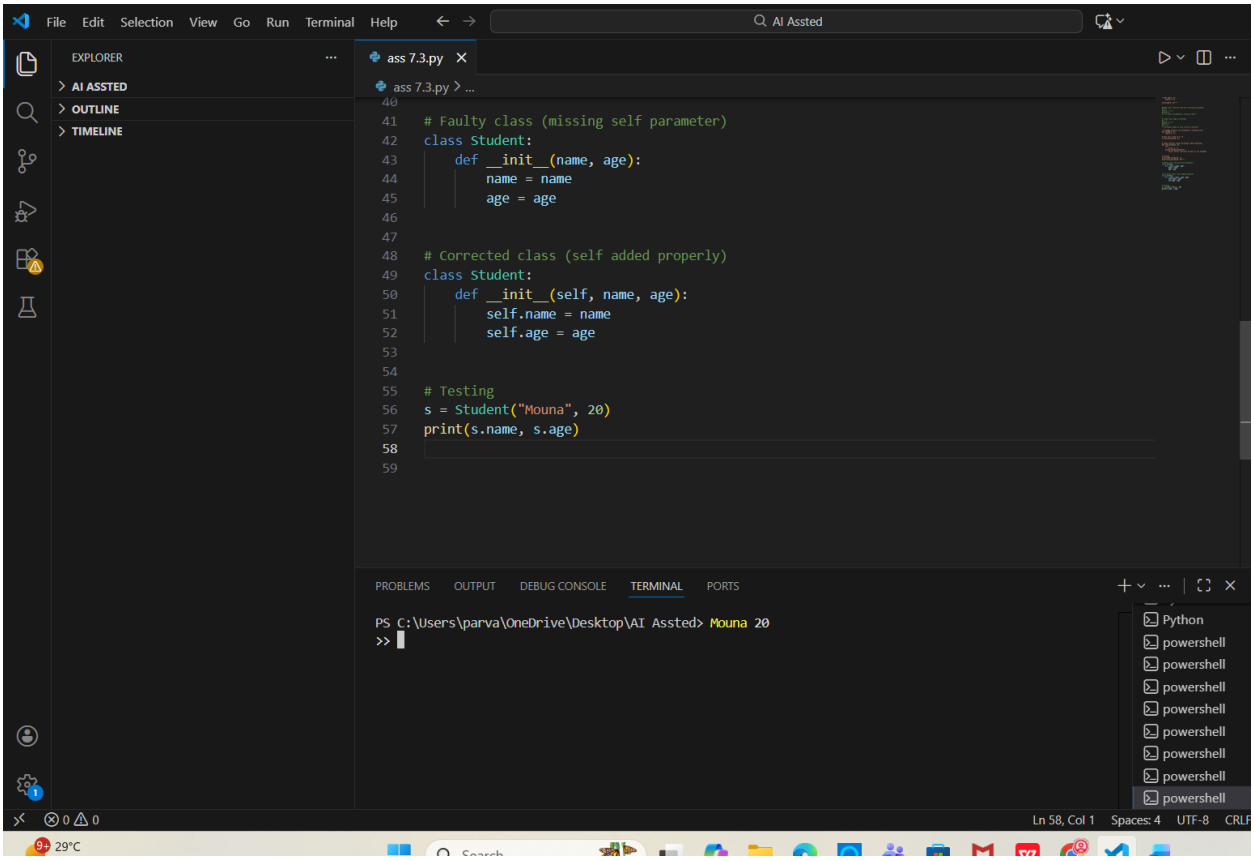
• Understand why self is required

Expected Output

• Corrected __init__() method

• Proper use of self in class definition

• AI explanation of object-oriented error



Task 5: Resolving Index Errors in Lists

Scenario

A program crashes when accessing an invalid index in a list.

Requirements

• Provide code that accesses an out-of-range list index

• Use AI to identify the Index Error

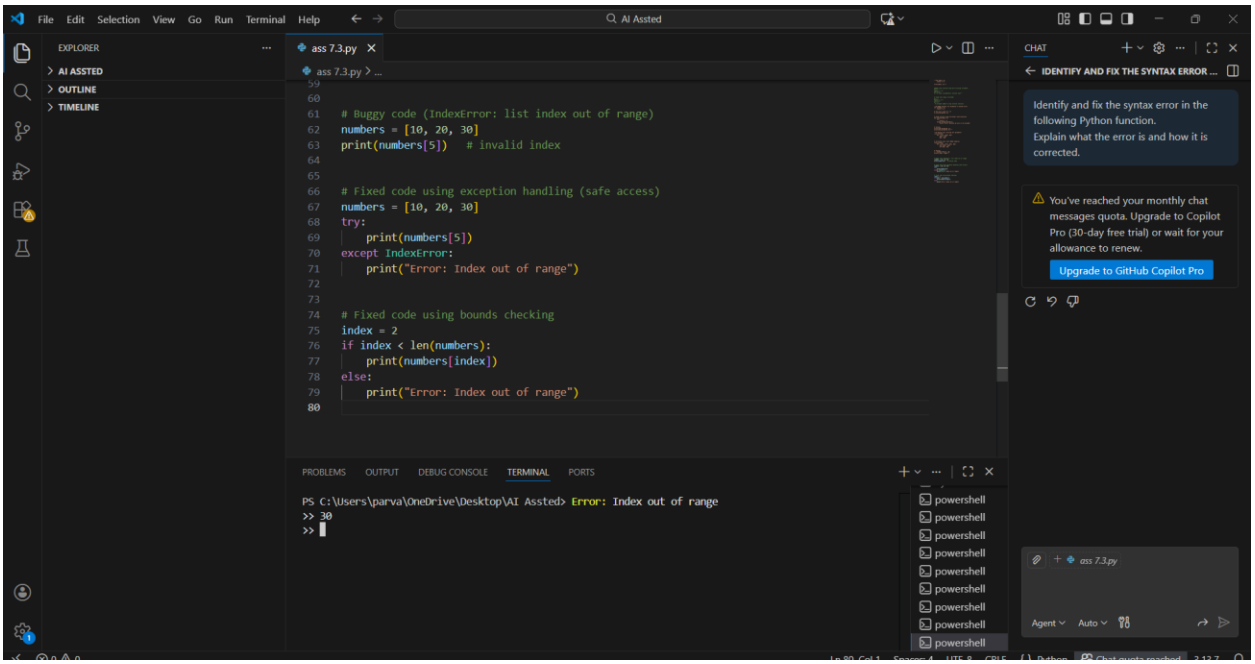
• Let AI suggest safe access methods

• Apply bounds checking or exception handling

Expected Output

• Index error resolved

• Safe list access logic implemented


File Edit Selection View Go Run Terminal Help
AI Assted
EXPLORER
AI ASSTED
OUTLINE
TIMELINE
ass 7.3.py
ass 7.3.py > ...
# Buggy code (IndexError: list index out of range)
numbers = [10, 20, 30]
print(numbers[5]) # invalid index
# Fixed code using exception handling (safe access)
numbers = [10, 20, 30]
try:
print(numbers[5])
except IndexError:
print("Error: Index out of range")
# Fixed code using bounds checking
index = 2
if index < len(numbers):
print(numbers[index])
else:
print("Error: Index out of range")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\parva\OneDrive\Desktop\AI Assted> Error: Index out of range
>> 30
>>
powershell
CHAT
IDENTIFY AND FIX THE SYNTAX ERROR ...
Identify and fix the syntax error in the following Python function.
Explain what the error is and how it is corrected.
You've reached your monthly chat messages quota. Upgrade to Copilot Pro (30-day free trial) or wait for your allowance to renew.
Upgrade to GitHub Copilot Pro
ass 7.3.py
Agent
Auto