AI Assisted Coding-Lab 7.3

Shivani Pambidi

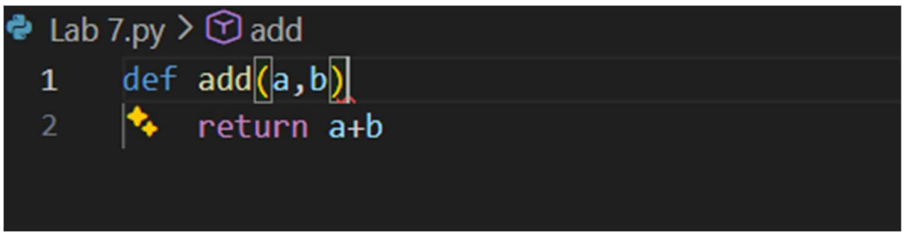
2303A52172

Batch – 41

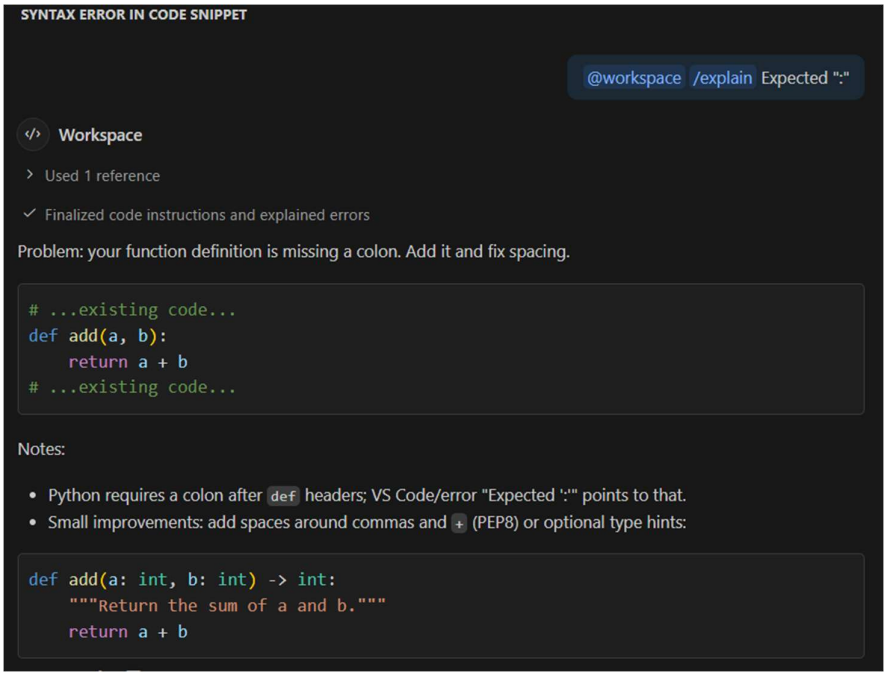
Question 1:

TASK 1 : Fixing Syntax Errors

CODE:



EXPLAINATION:



JUSTIFICATION:

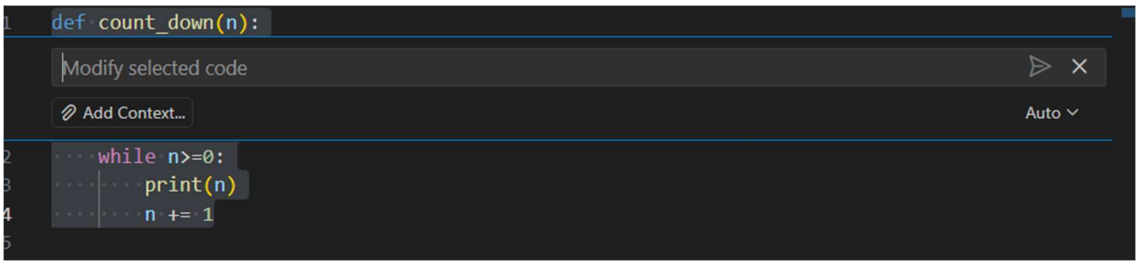
The function had a syntax error because the colon was missing after the function definition. Python requires a colon to start the function body. An AI

tool identified this error and fixed it by adding the missing colon. After the correction, the function executed properly and the syntax error was resolved.

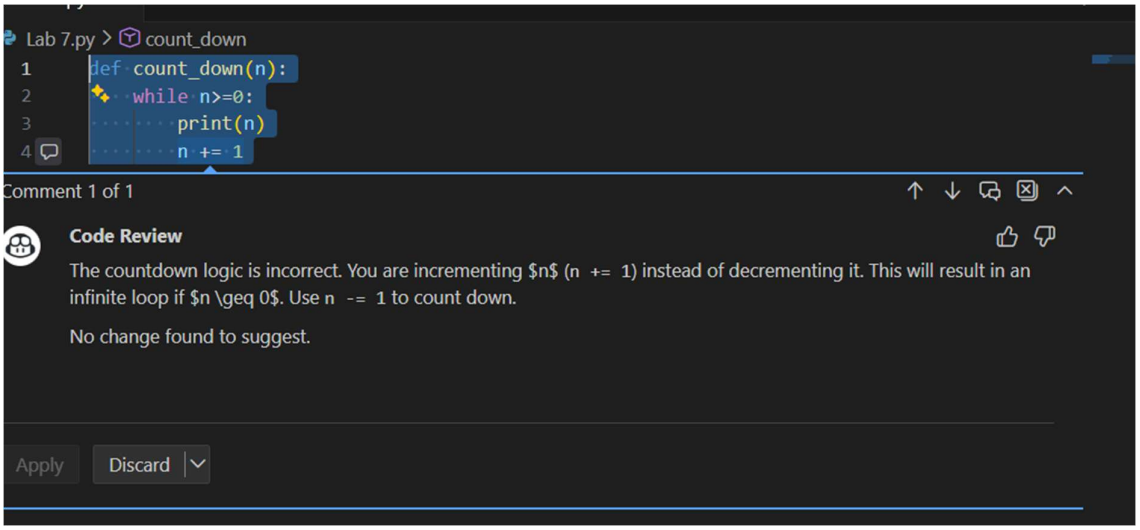
Question 2:

TASK 2: Debugging Logic Errors in Loops

MODIFY:



REVIEW:



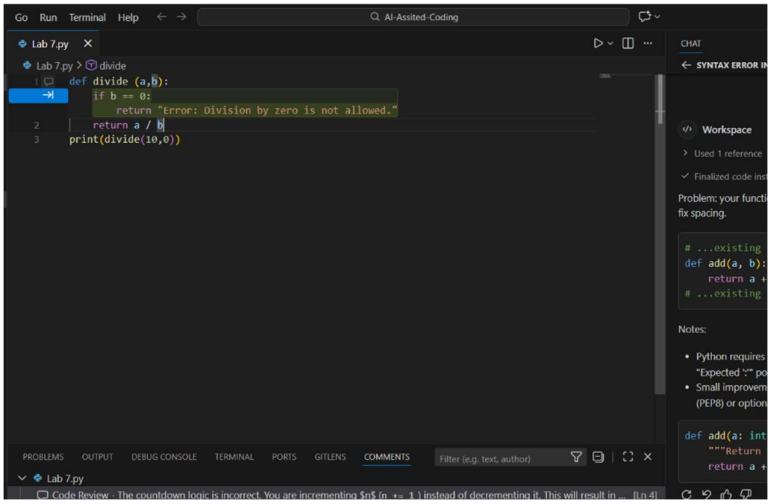
JUSTIFICATION:

The loop was running infinitely because the value of n was increasing instead of decreasing. The condition while n >= 0 always remained true since n never moved toward zero. An AI tool identified this logical mistake by analyzing the loop condition and update statement. The AI corrected the logic by changing the increment to a decrement so that n reduces in each iteration. After this fix, the loop stops correctly at zero and the infinite loop issue is resolved.

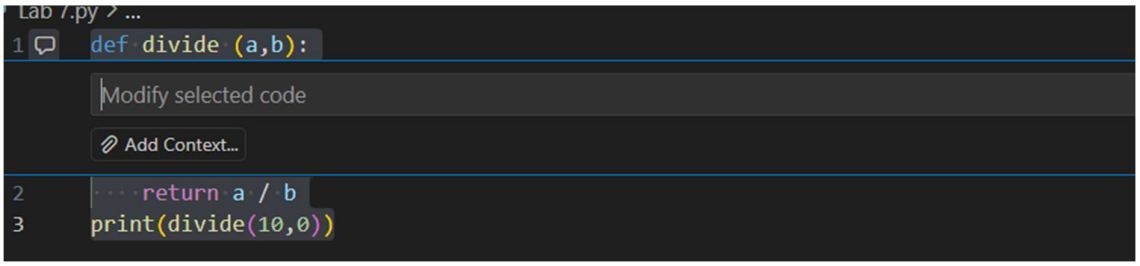
Question 3:

TASK 3 : Handling Runtime Errors (Division by Zero)

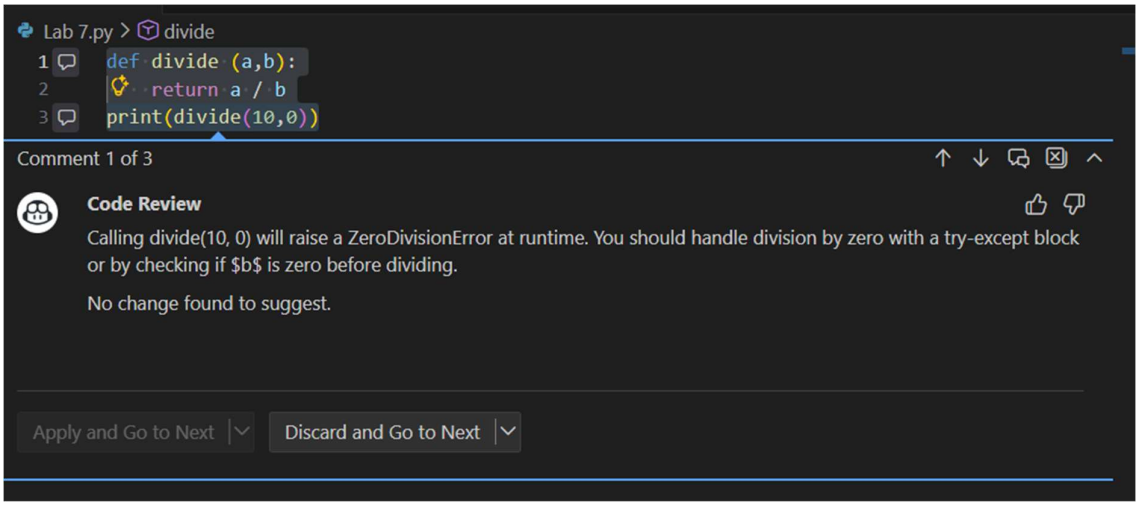
EXTRACT:

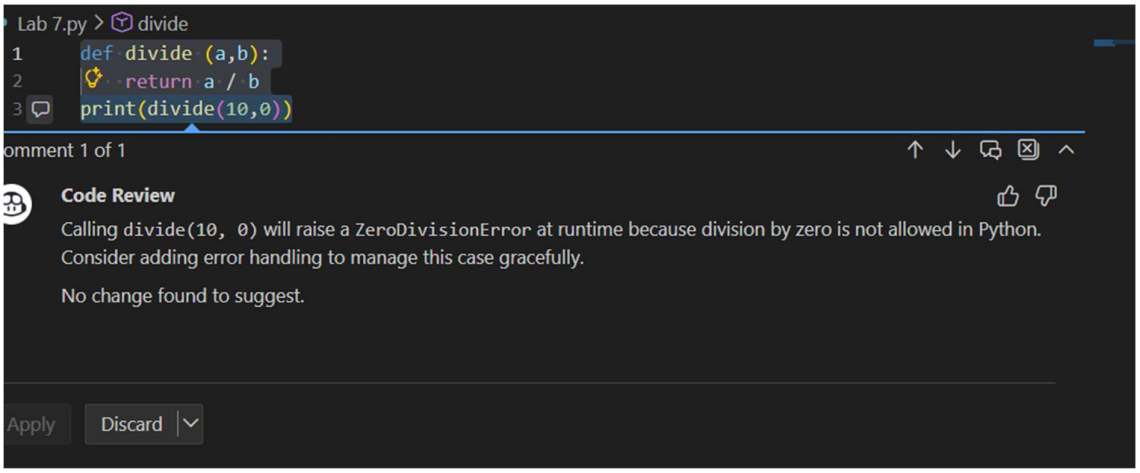


MODIFY:



REVIEW:





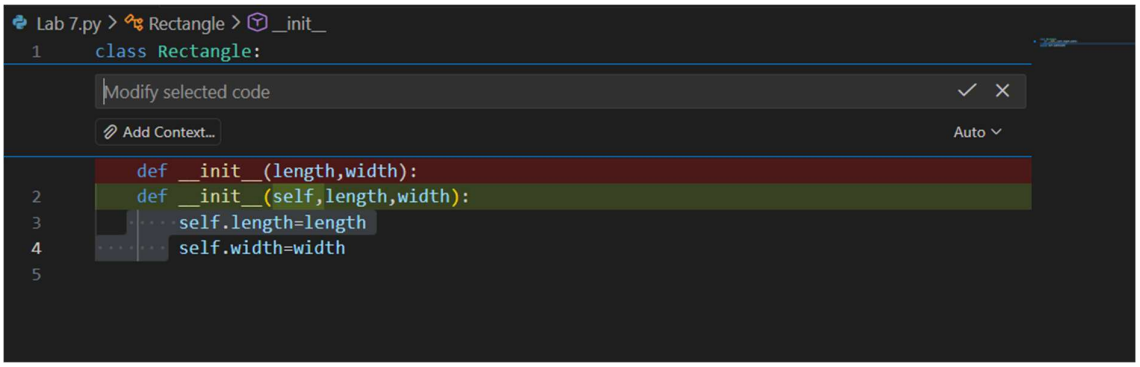
JUSTIFICATION:

The program crashes because it tries to divide a number by zero, which is not allowed in Python. An AI tool identifies this as a runtime error that occurs during execution. The AI fixes the issue by adding a try-except block to catch the division by zero error. With this handling, the program runs safely without crashing and shows a proper message instead.

Question 4:

TASK 4: Debugging Class Definition Errors

FIX:



EXPLAINATION:

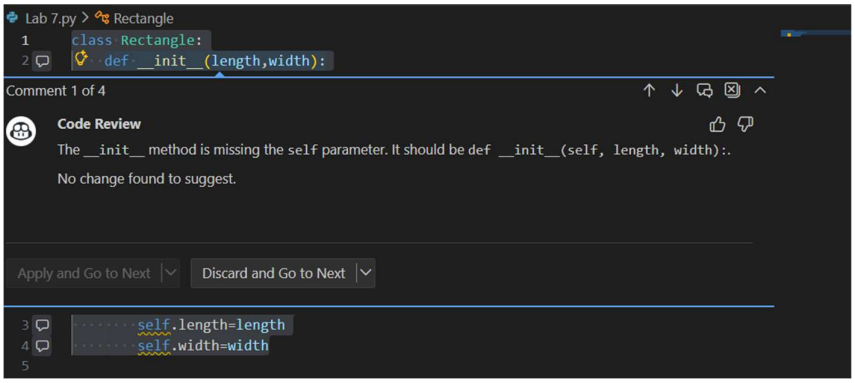
Problem: the instance method is missing the required first parameter (conventionally named `self`), so references to `self` in the body are undefined. Fix by adding `self` as the first parameter.

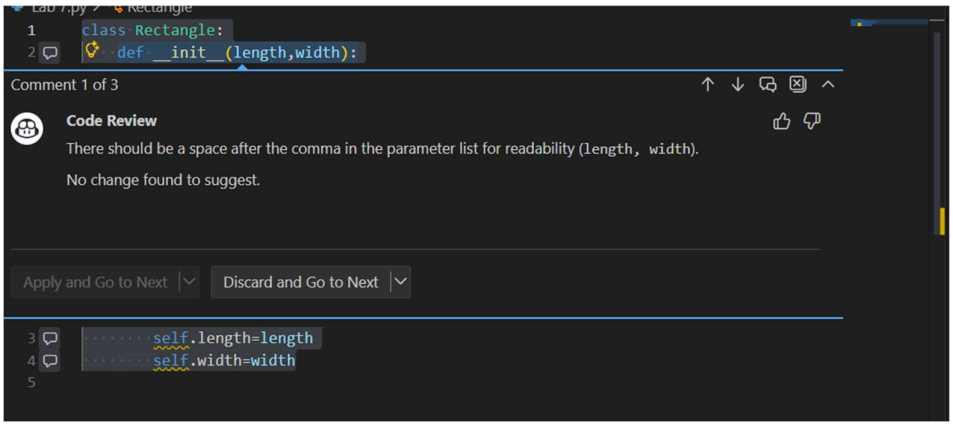
```
# ...existing code...
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width
# ...existing code...
```

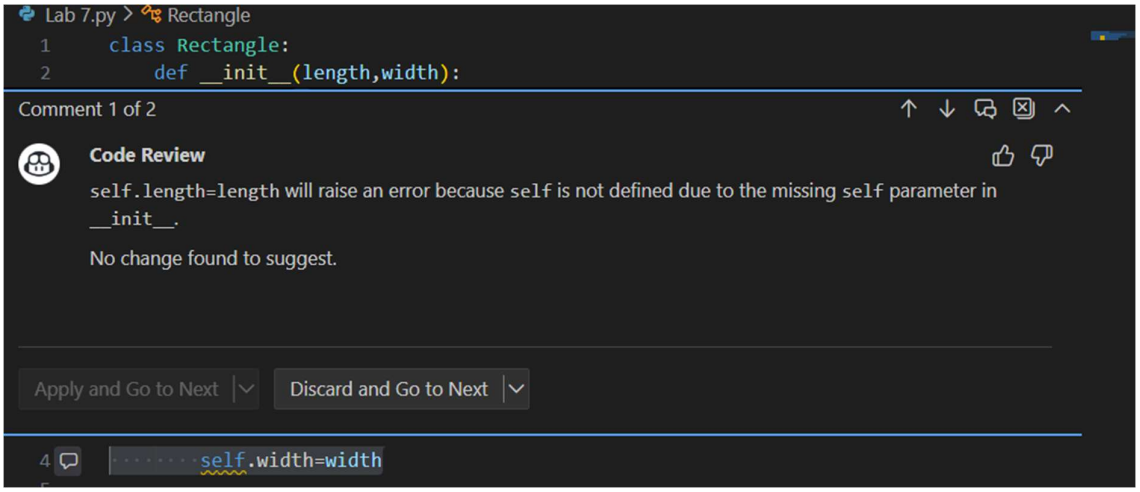
Notes:

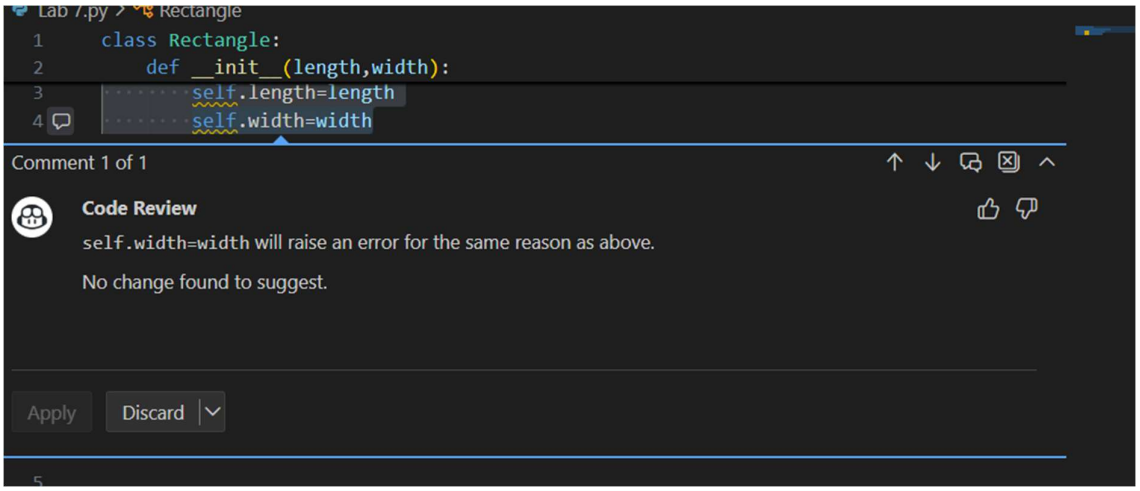
- Methods must accept the instance as the first parameter (use `self` by convention).
- Optional improvements: add type hints (`def __init__(self, length: float, width: float) -> None:`), validation, or helper methods like `area()` or `__repr__()`.

REVIEW:









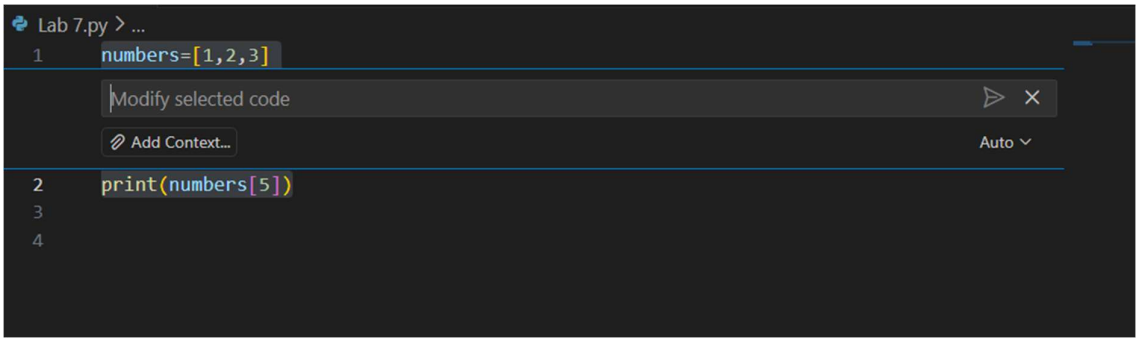
JUSTIFICATION:

The class had an error because the __init__ method was missing the self parameter. In Python, self is required to refer to the current object inside the class. An AI tool detected this object-oriented mistake and corrected the constructor by adding self as the first parameter. After the fix, object attributes are assigned correctly and the class works as expected.

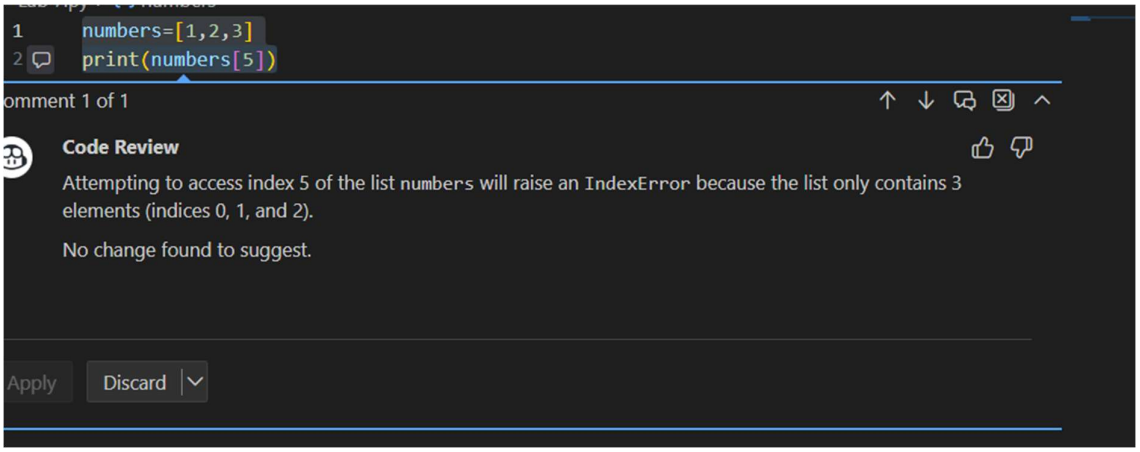
Question 5:

TASK 5: Resolving Index Errors in Lists

MODIFY:



REVIEW:



JUSTIFICATION:

The program crashes because it tries to access an index that does not exist in the list. An AI tool identifies this as an IndexError caused by an out-of-range index value. The AI suggests fixing it by checking the list length or using exception handling before accessing the element. After applying safe access logic, the program runs without crashing and the index error is resolved.