LAB ASSIGNMENT-7.3

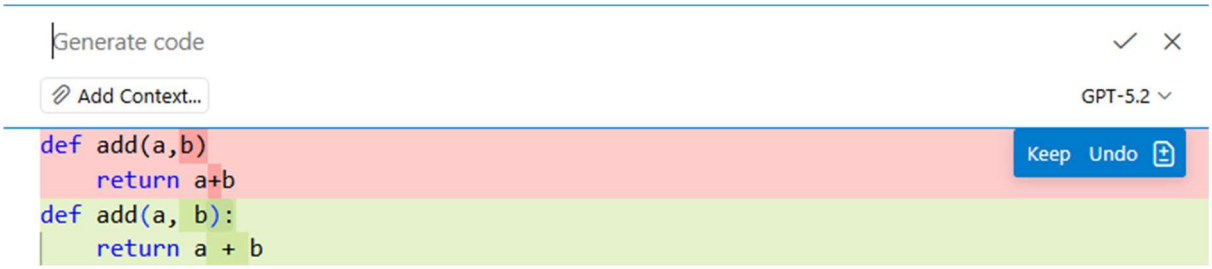
NAME: SINDHU KATAKAM

ENROLLEMNT NUMBER: 2303A52481

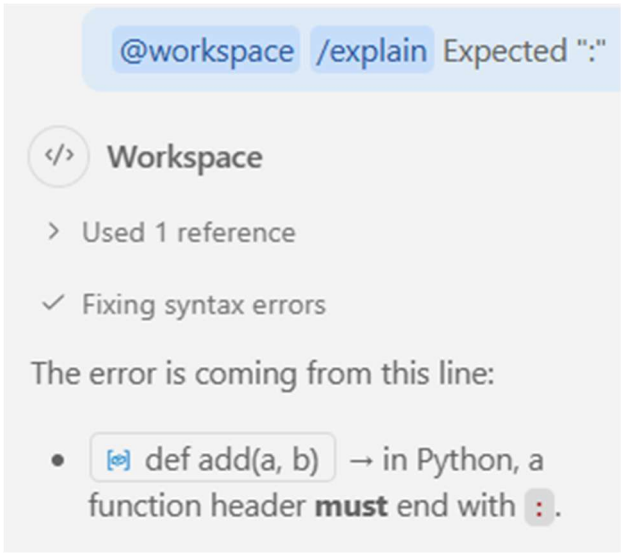
BATCH: 41

Task 1: Fixing Syntax Errors

FIX:



EXPLAIN:



JUSTIFICATION:

The syntax error happened because the function definition line did not end with a colon. In Python, a colon tells the interpreter that an indented block of code will follow. Adding the colon fixes the structure and allows the function to run correctly.

Task 2: Debugging Logic Errors in Loops

FIX:

```
def count_down(n):
    while n>=0:
        print(n)
        n+=1
        n -= 1
print(count_down(5))
```

Keep Undo

REVIEW:

Code Review

The function increments n instead of decrementing it, causing an infinite loop. For a countdown, change n+=1 to n-=1.

Suggested change:

```
-           n+=1
+           n-=1
```

JUSTIFICATION:

The infinite loop happened because the loop variable was not incremented or decremented properly. Updating the variable correctly ensures the loop condition will eventually become false, preventing infinite execution.

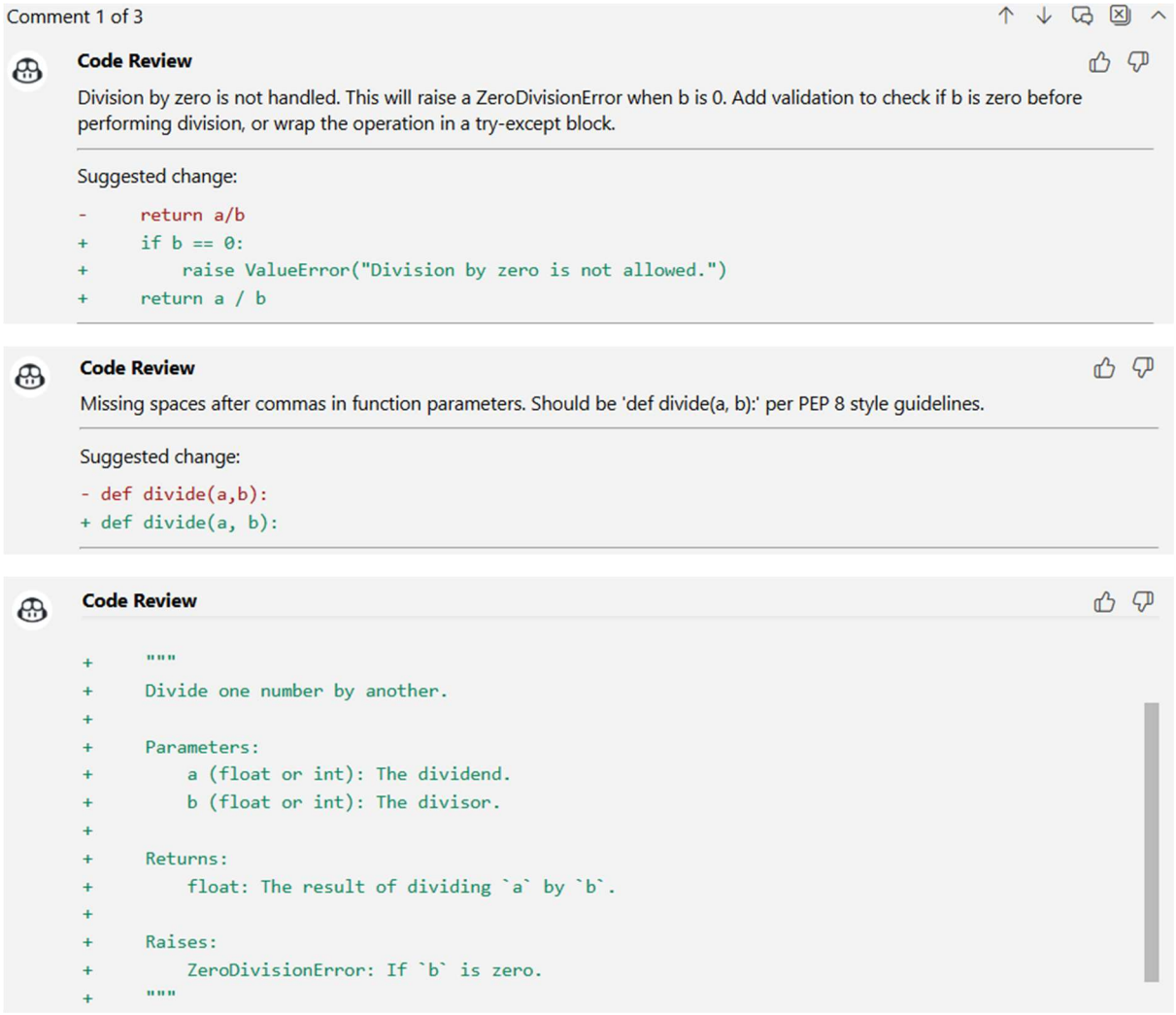
Task 3: Handling Runtime Errors (Division by Zero)

FIX:

```
def divide(a,b):
    return a/b
print(divide(10,0))
def divide(a, b):
    try:
        return a / b
    except ZeroDivisionError:
        return None

result = divide(10, 0)
print("Cannot divide by zero" if result is None else result)
```

REVIEW:

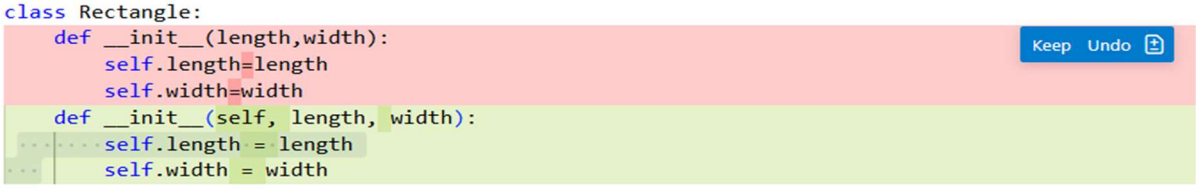


JUSTIFICATION:

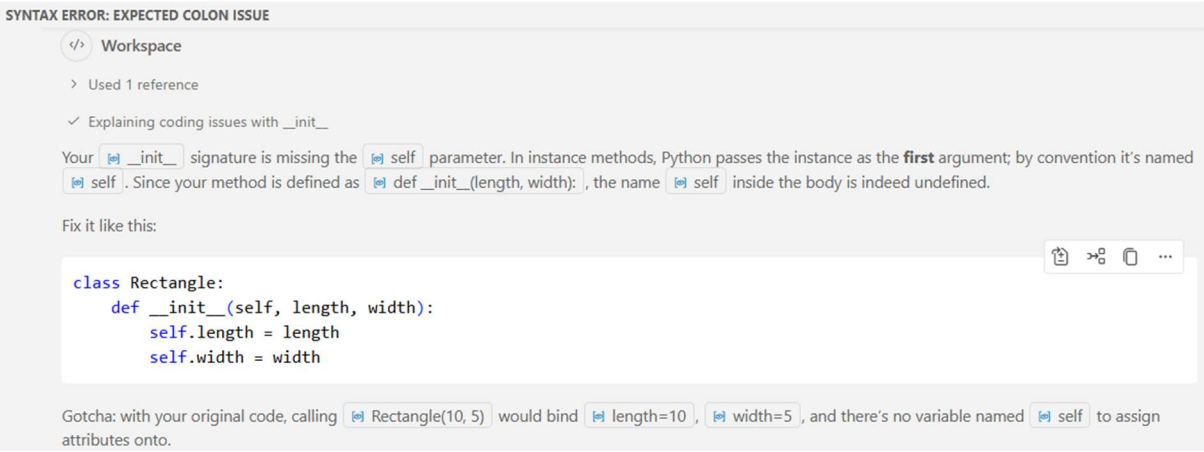
In this task, a function caused an error because a number was divided by zero. This error happens while the program is running, so it is called a runtime error. To fix this problem, a try-except block was used to handle the error safely. The program now shows an error message instead of crashing. This makes the program more reliable and user-friendly.

Task 4: Debugging Class Definition Errors

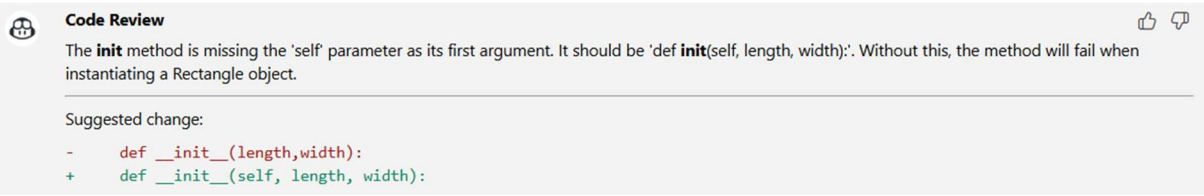
MODIFY:



EXPLAIN:



REVIEW:

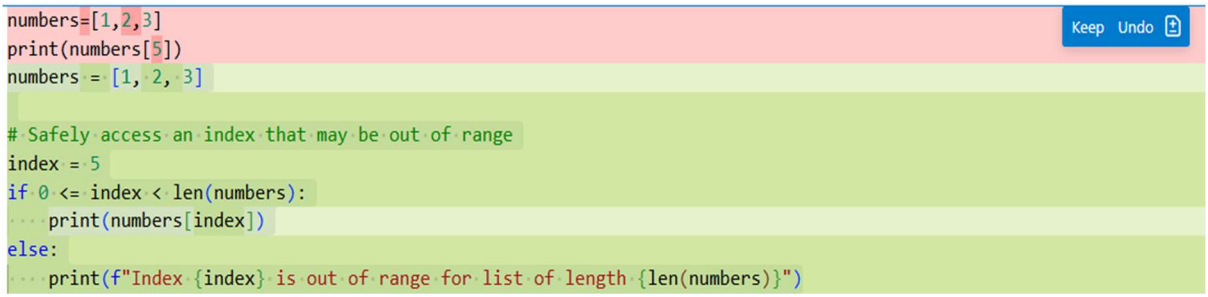


JUSTIFICATION:

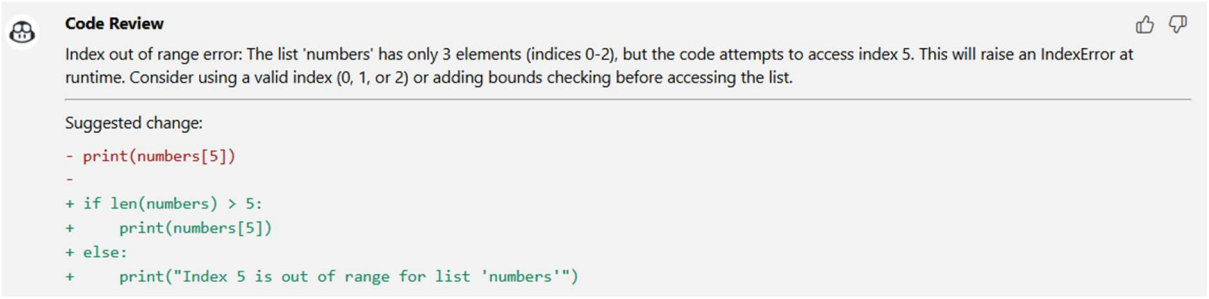
In this task, the class constructor was written incorrectly because the self parameter was missing. The self keyword is important because it allows the object to store and access its own data. Without self, the class cannot work properly. After adding self, the class works correctly and can create objects. This helps in understanding how object-oriented programming works in Python.

Task 5: Resolving Index Errors in Lists

MODIFY:



REVIEW:



JUSTIFICATION:

In this task, an index error occurred when trying to access a list element that does not exist. This happens when the index number is outside the range of the list. The issue was fixed by using exception handling or checking the list length before accessing the index. This prevents the program from crashing. It helps in writing safer and more stable code.