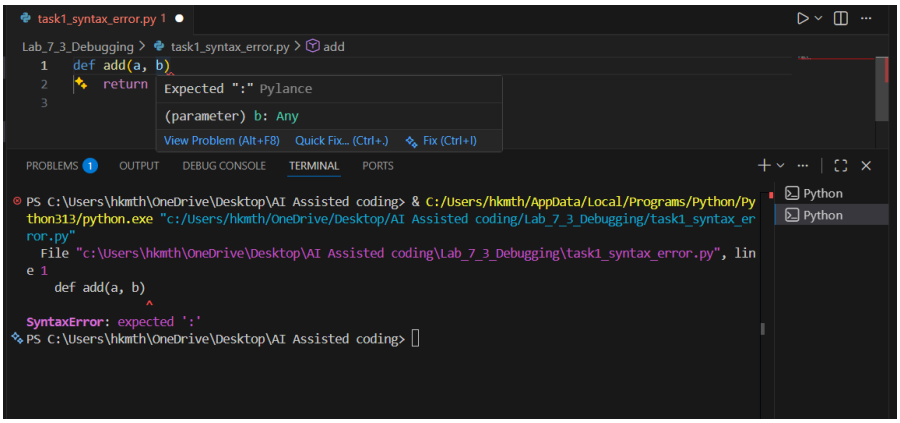
ASSIGNMENT – 7.3

Thumma Hasini | 2303A52076 | Batch – 37

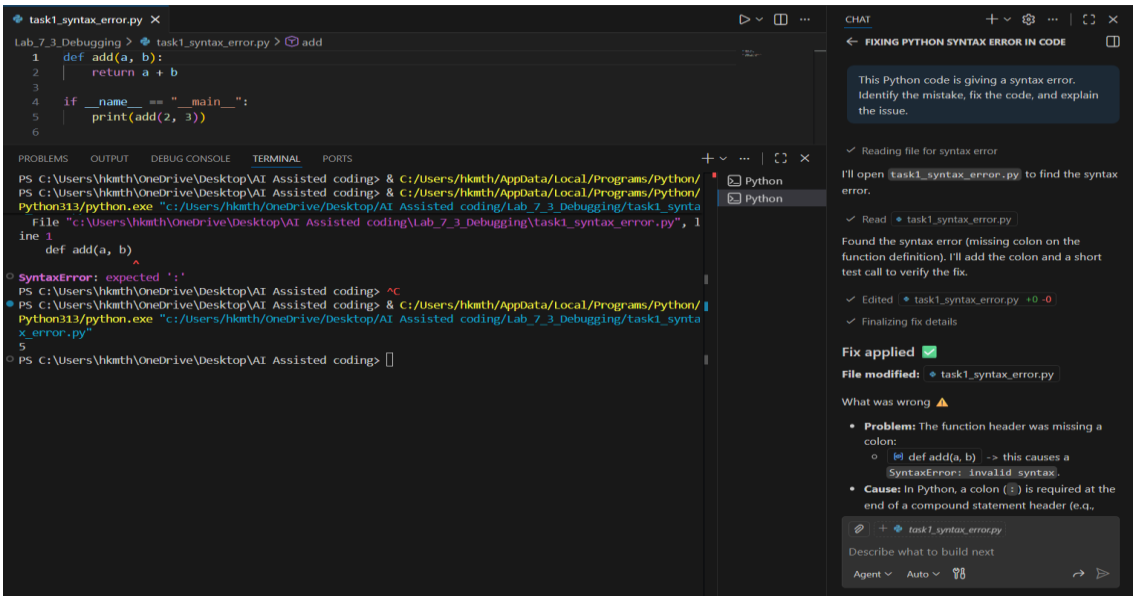
Task 1: Fixing Syntax Errors

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

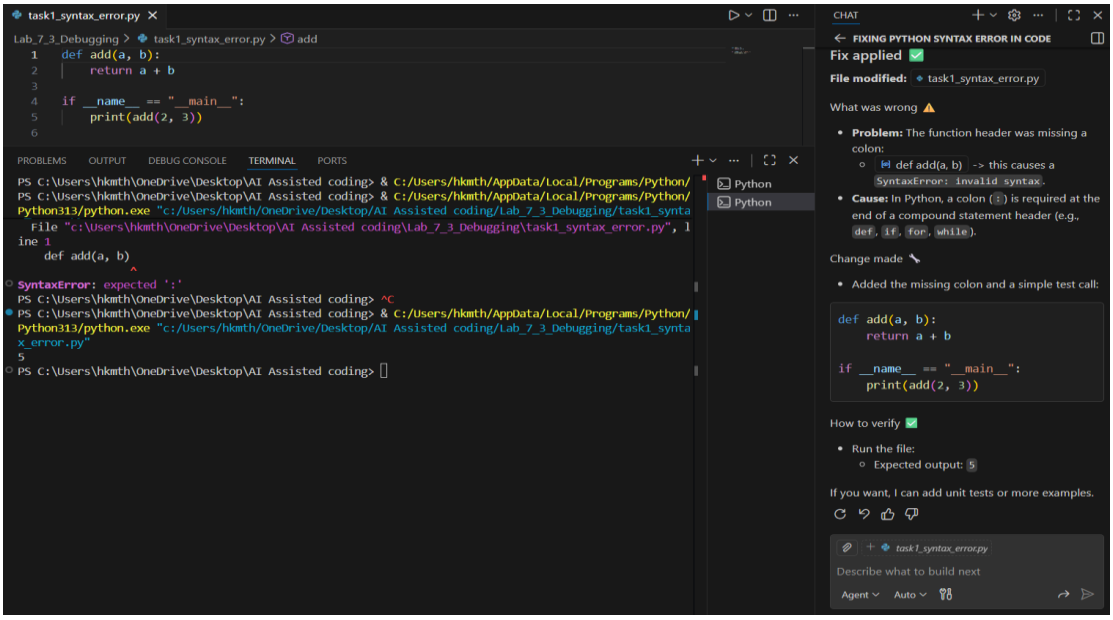
```python
def add(a, b)
    return a + b
```



Prompt: This Python code is giving a syntax error. Identify the mistake, fix the code, and explain the issue.



.

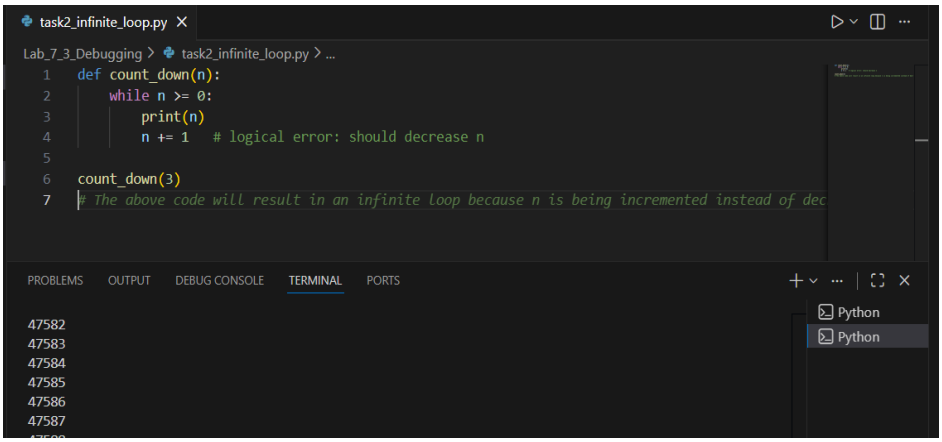


Explanation: The buggy Python code is executed in VS Code to observe the syntax error. The error is then fixed using AI suggestions, and the corrected code runs successfully.

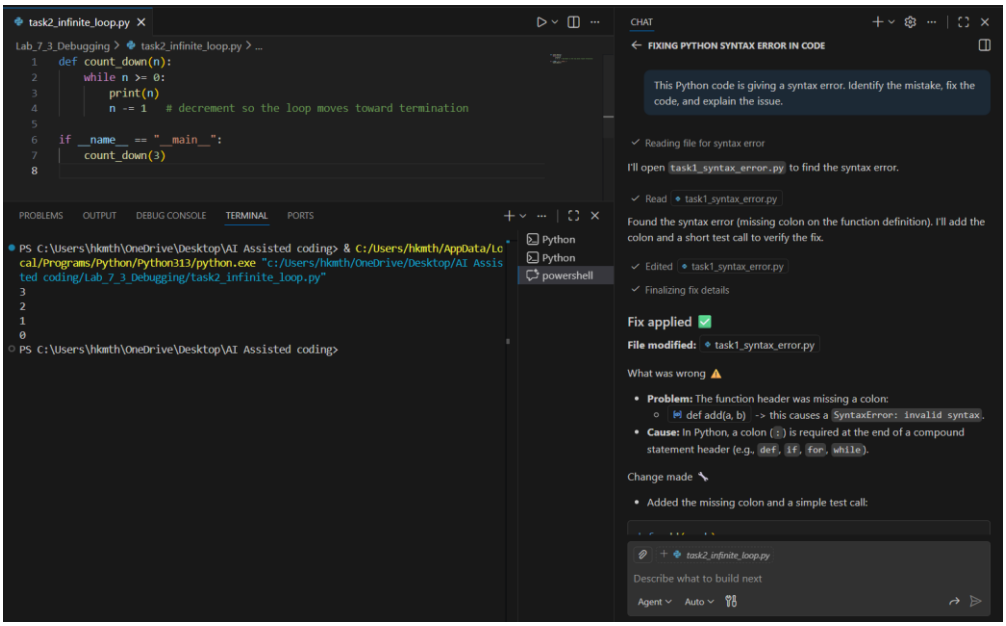
Task 2: Debugging Logic Errors in Loops

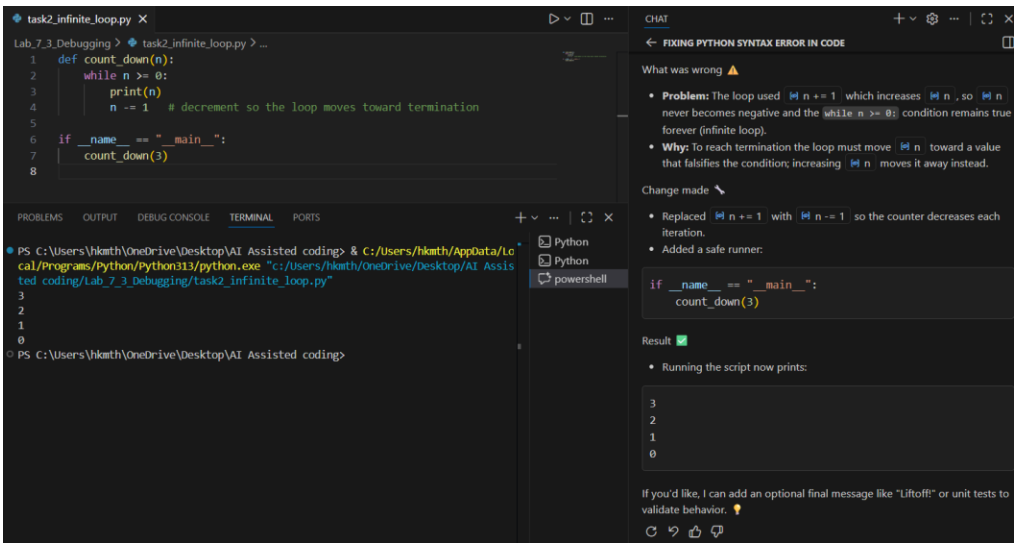
You are debugging a loop that runs infinitely due to a logical mistake.

```python
def count_down(n):
    while n >= 0:
        print(n)
        n += 1  # Should be n -= 1
```



Prompt : The following Python function runs infinitely due to a logic error in the loop. Identify the mistake, explain why the loop does not terminate, and correct the code so that the countdown works properly.





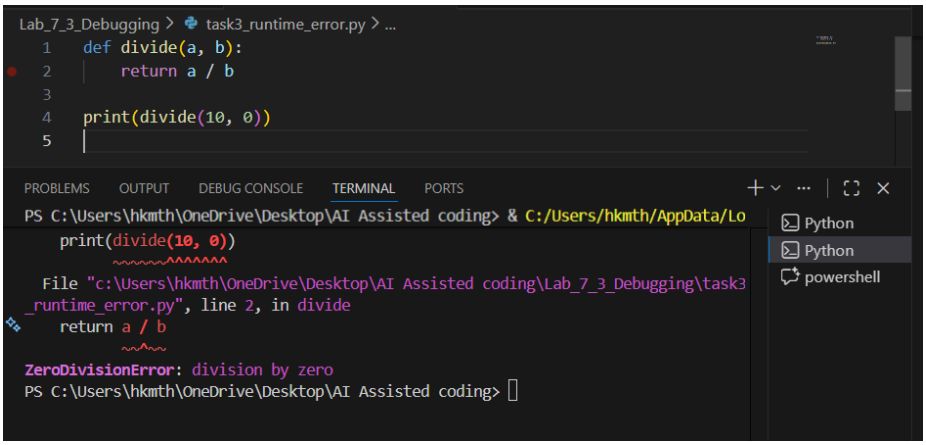
Explanation: The loop runs infinitely because the value of n is increased instead of decreased, so the loop condition always remains true. The AI identified this logical error and corrected the update statement. After fixing it, the loop properly counts down and terminates.

Task 3: Handling Runtime Errors (Division by Zero)

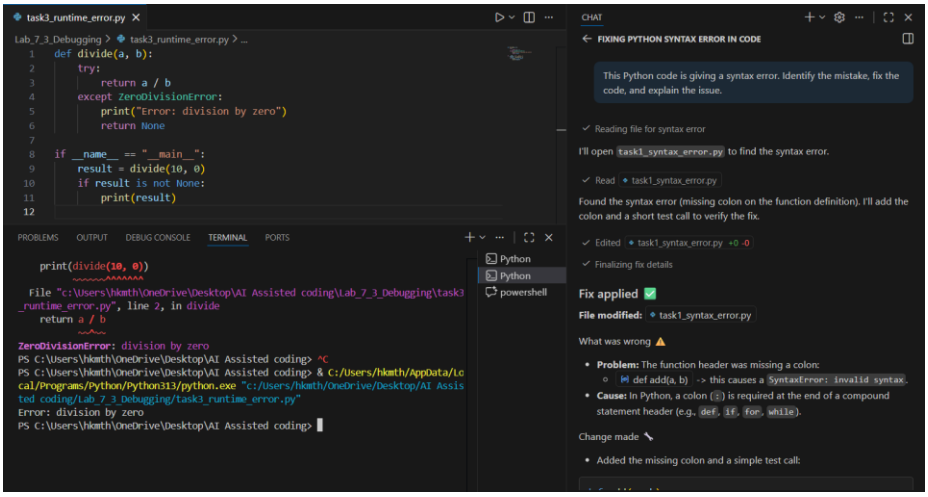
A Python function crashes during execution due to a division by zero error.

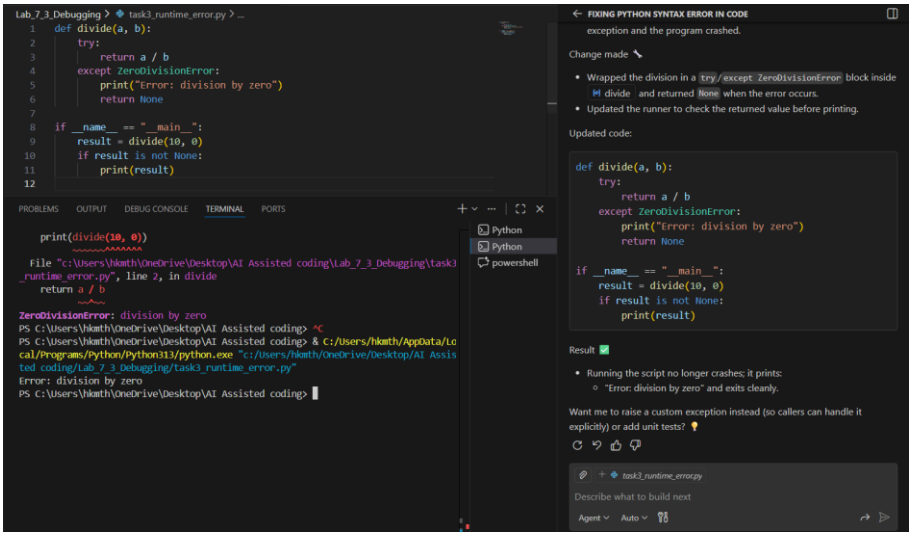
```python
# Debug the following code
def divide(a, b):
    return a / b

print(divide(10, 0))
```



Prompt: The following Python function causes a runtime error during execution. Identify the error, explain why it occurs, and modify the code using try-except to handle the error safely.



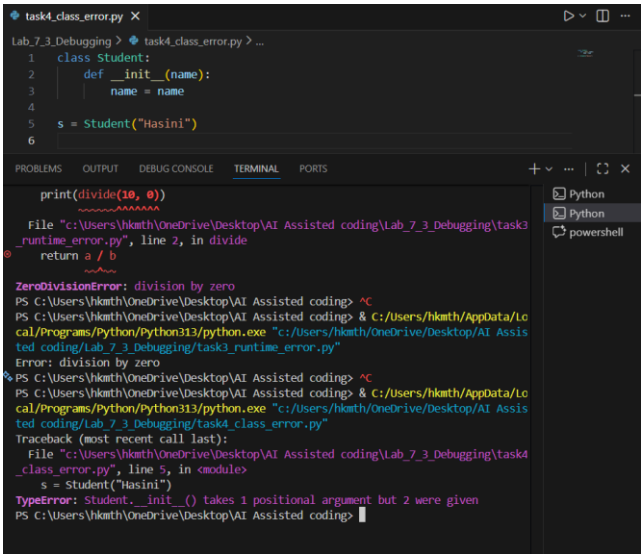


Explanation: The program crashes at runtime because it attempts to divide a number by zero. This results in a Zero Division Error, which is a runtime error.

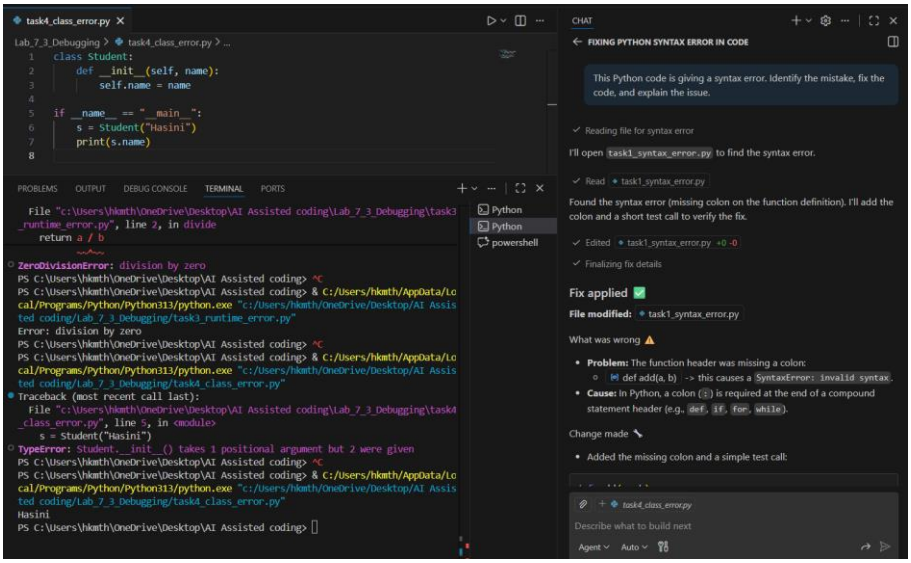
Task 4: Debugging Class Definition Errors

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

```python
class Rectangle:
    def __init__(length, width):
        self.length = length
        self.width = width
```



Prompt: The following Python class is giving an error during object creation. Identify the issue in the constructor, correct the class definition, and explain why the error occurs.

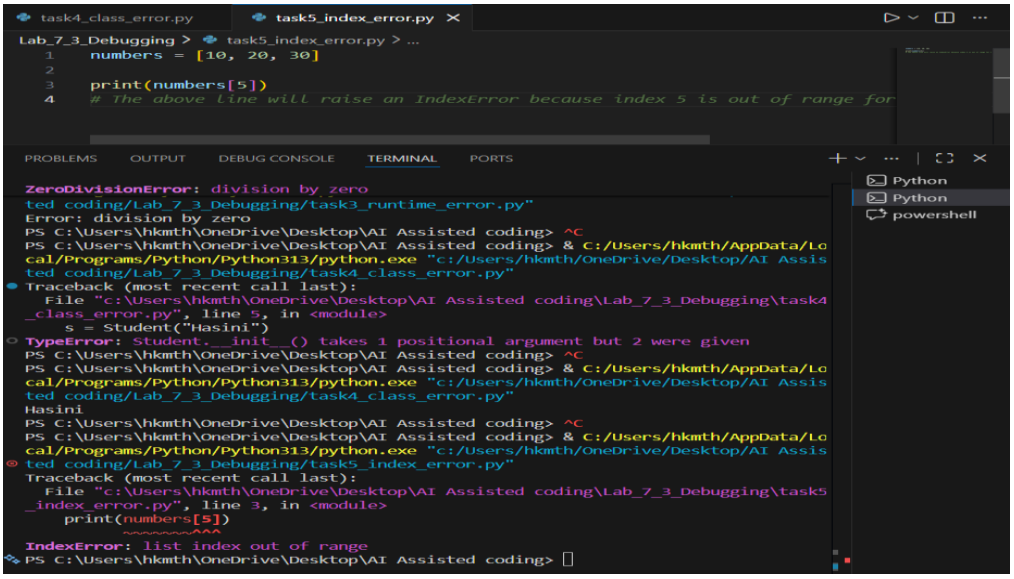


Explanation: In this task, a runtime error occurs because the self parameter is missing in the class constructor. Python automatically passes the object reference to methods, and without self, the arguments do not match. The AI fixes the issue by adding self to the constructor definition.

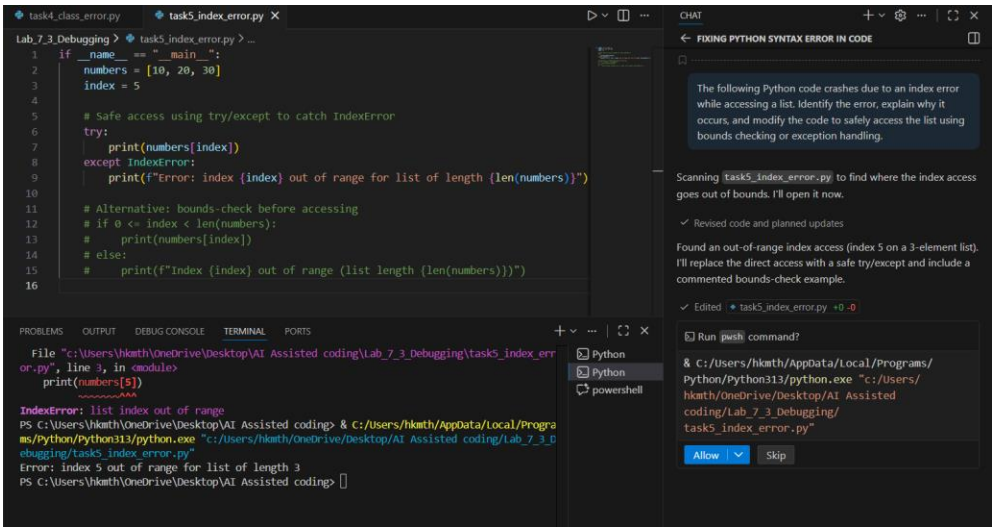
Task 5: Resolving Index Errors in Lists

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

```python
numbers = [1, 2, 3]
print(numbers[5])
```



Prompt: The following Python code crashes due to an index error while accessing a list. Identify the error, explain why it occurs, and modify the code to safely access the list using bounds checking or exception handling.



Explanation: In this task, the program crashes because it tries to access a list index that does not exist. This causes an Index Error at runtime. The AI fixes the issue by handling the error safely, preventing the program from crashing.